IPsec – Encrypted and secure IP

Chapter 20

IPsec

- Overview:
  - Two operating modes: Tunnel and Transport Mode
  - Two protocols: AH and ESP
  - And one Key Exchange protocol: IKE

- Standards [2005]:

  RFC 4301: Overview
  RFC 4302: Authentication Header (AH)
  RFC 4303: Encapsulating security payload (ESP)
  RFC 7296: Key management (IKEv2) [2014]

- Updates and additions (new ciphers and algorithms) in newer RFCs

IPsec objectives

- Encryption of traffic at IP level
  - Transparent for transport layer (TCP, UDP)
  - Independent of network technology
  - De-facto standard for site-to-site VPNs
  - Supported in Windows 2k and later
- (Almost) mandatory in IPv6, optional in IPv4
  - IPsec is not mandatory, but "IPv6 should support the IPsec architecture"
  - Sometimes other techniques may be more appropriate (TLS, SSH, …)
- Application examples:
  - Network connectivity over the Internet (*site → site*)
  - Secure remote access (*user → site* and *user → server*)
  - *Server → server* traffic encryption
- Functionality in IPsec:
  - Access Control, Message integrity, Data origin authentication, Rejection of replayed packets, Confidentiality (encryption)

IPsec: Site to site and user to site

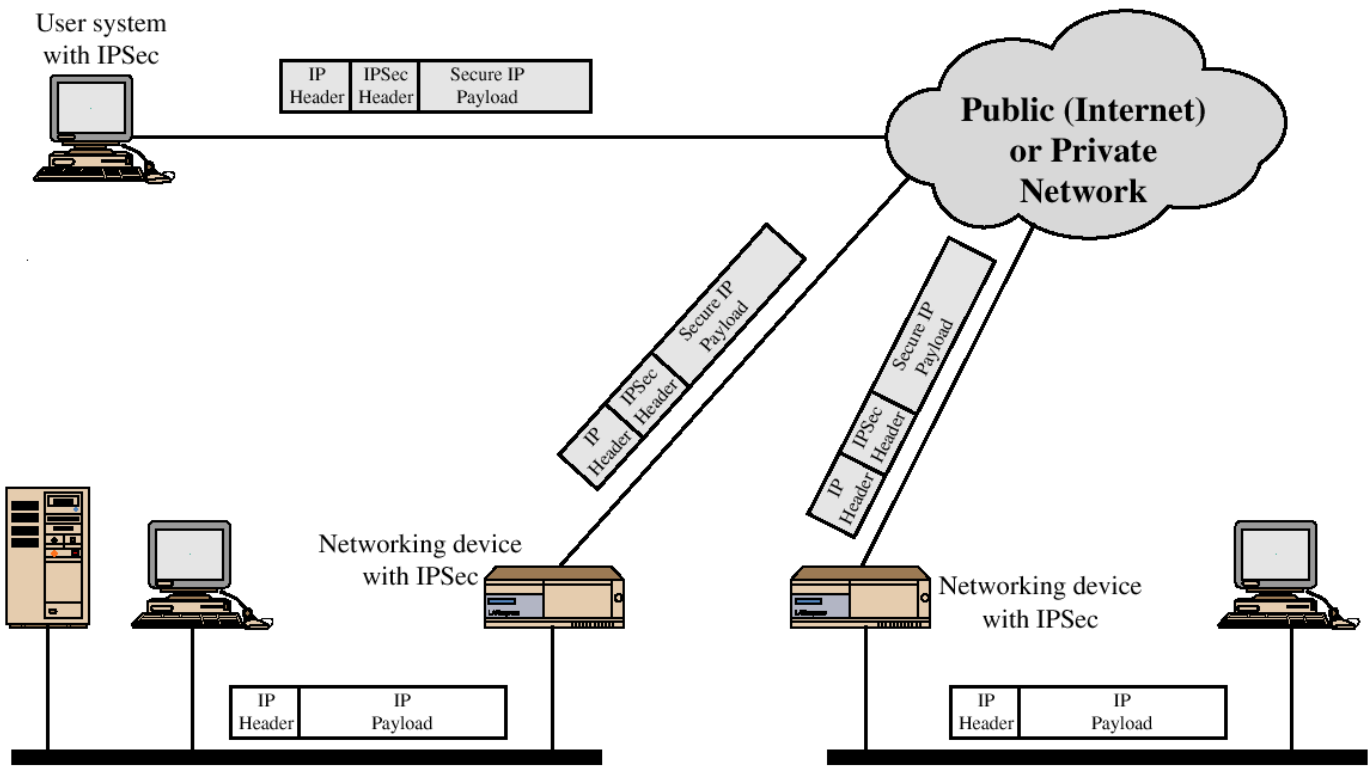
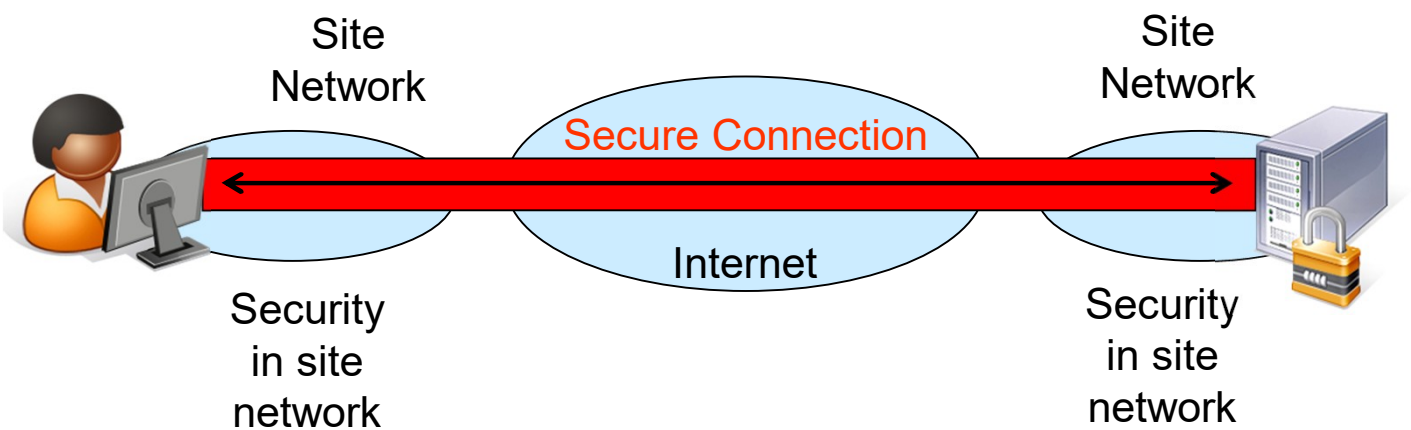


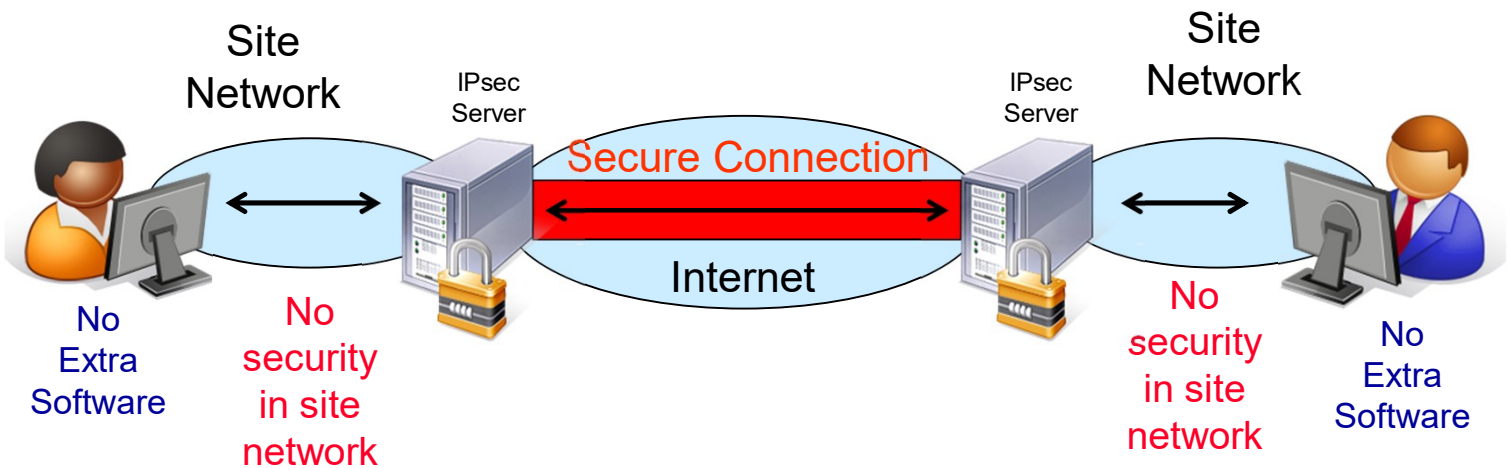
Fig. 20.8

Transport Mode – end-to-end



Offers end-to-end encryption
Often used for remote access
Drawback: End-devices must implement IPsec

Tunnel Mode – site-to-site



Offers site-to-site encryption (firewall to firewall)
Used to implement Virtual Private Networks (VPN)

Tunnel mode – the client's view

Running traceroute from any remote location when having an IPsec (VPN) tunnel active to Chalmers:

> **tracert www.dn.se**

Tracing route to [217.114.89.134] over a maximum of 30 hops:

1	63 ms	63 ms	67 ms	vpn-dialin-140-10.vpn.chalmers.se
2	64 ms	63 ms	62 ms	cth29a-itss-gw.chalmers.se
3	64 ms	63 ms	63 ms	core1-itss-gw.chalmers.se

…

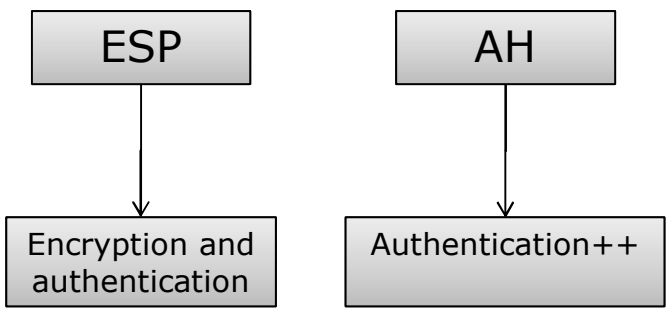
This is where the ICMP datagram is unpacked and meets the world!

IPsec protocols: (AH), ESP, IKE

Three protocols:

- Authentication Header (AH)
  - Provides support for data integrity
  - Uses HMAC to verify integrity
  - Does not encrypt messages – possible reasons: political and (minor) speed gain
- Encapsulating Security Payload (ESP)
  - Provides support for data integrity and message confidentiality
  - Algorithms can be different in different directions
- Internet Key exchange protocol (IKE)
  - Authenticates parties
  - Negotiates tunnel capabilities between two peers

AH and ESP



Auth++ also covers orig. IP header (e.g. source address) in transport mode

	ESP	AH
Access control	X	X
Data integrity	X	X
Packet authentication	X	X++
Replay protection	X	X
Confidentiality	X	

Table 20.2

- Authentication in ESP was not mandatory in early implementations – opened up for modification attacks
- AH is mainly used in equipment with low processing power where no encryption is needed
- Authentication++ normally not needed, see coming slides
- ESP performs Encrypt-then-MAC

ESP in Transport and Tunnel mode

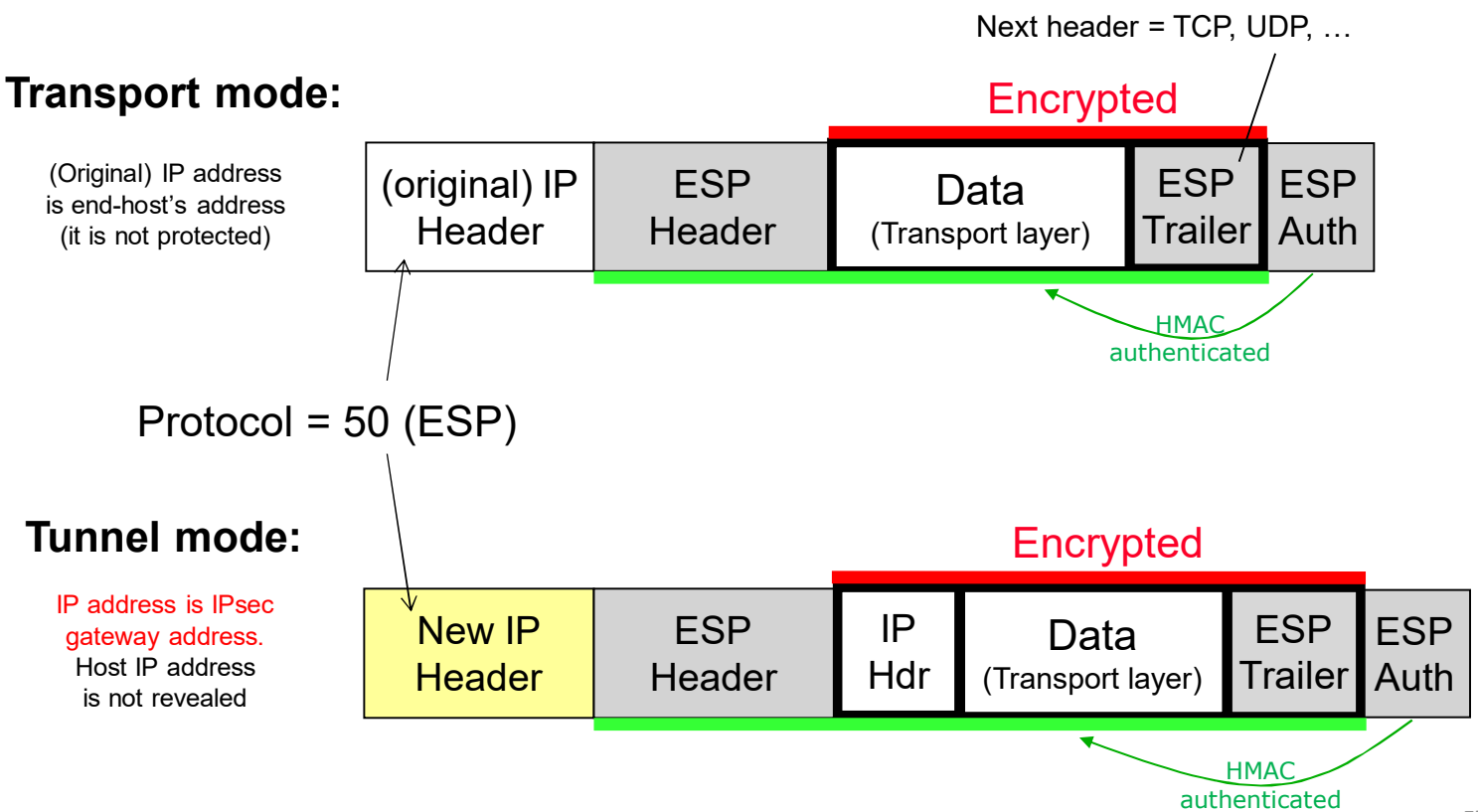


Fig. 20.6

The ESP packet

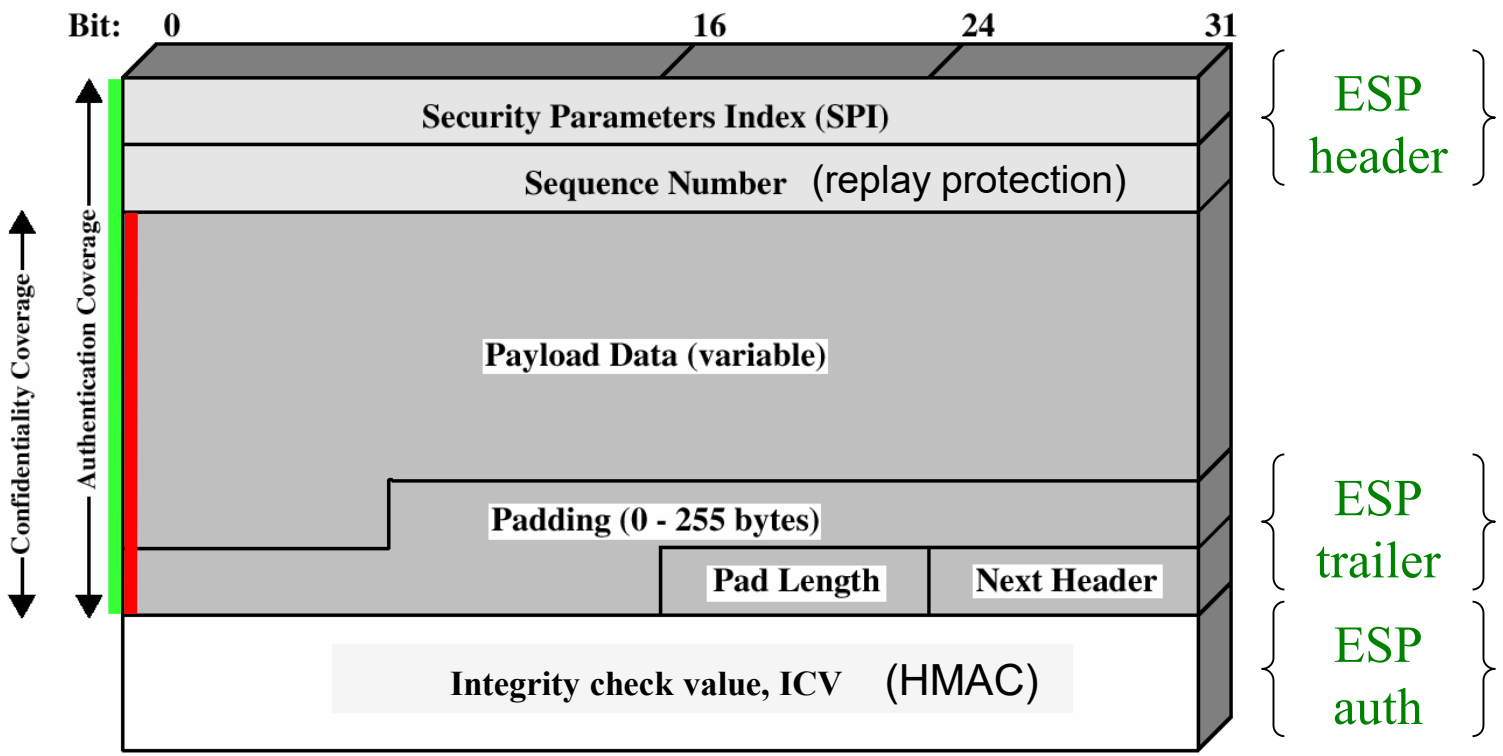


Fig. 20.4a

Using ESP

- Padding makes sure “Next header” is properly aligned
  - Necessary if cipher requires blocks of a certain size
  - Random padding supported
- The ICV does not cover the IP header in transport mode
  - ICV is a keyed HMAC
  - But changing the IP header just means it will not reach the destination
- Sequence numbers protect against duplicates and replays
  - 32-bit number, when exhausted: negotiate a new SA (see later)
- Receiver has a window of acceptable datagram numbers
  - IP may cause out-of-order delivery , default size = 64 packets
  - Duplicate numbers and numbers below the window are discarded

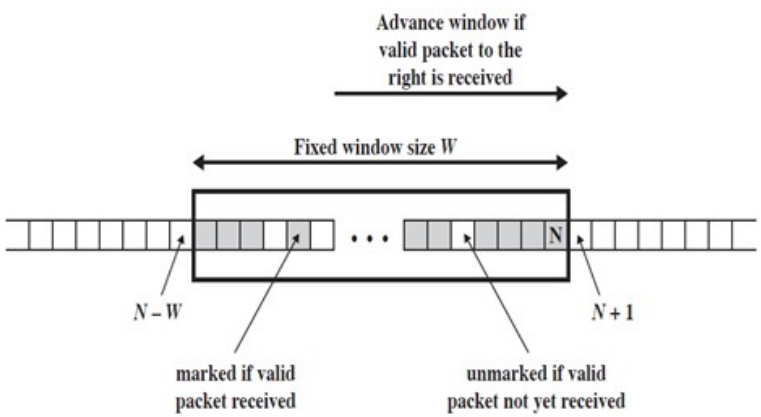
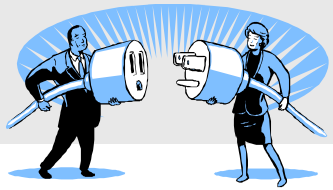


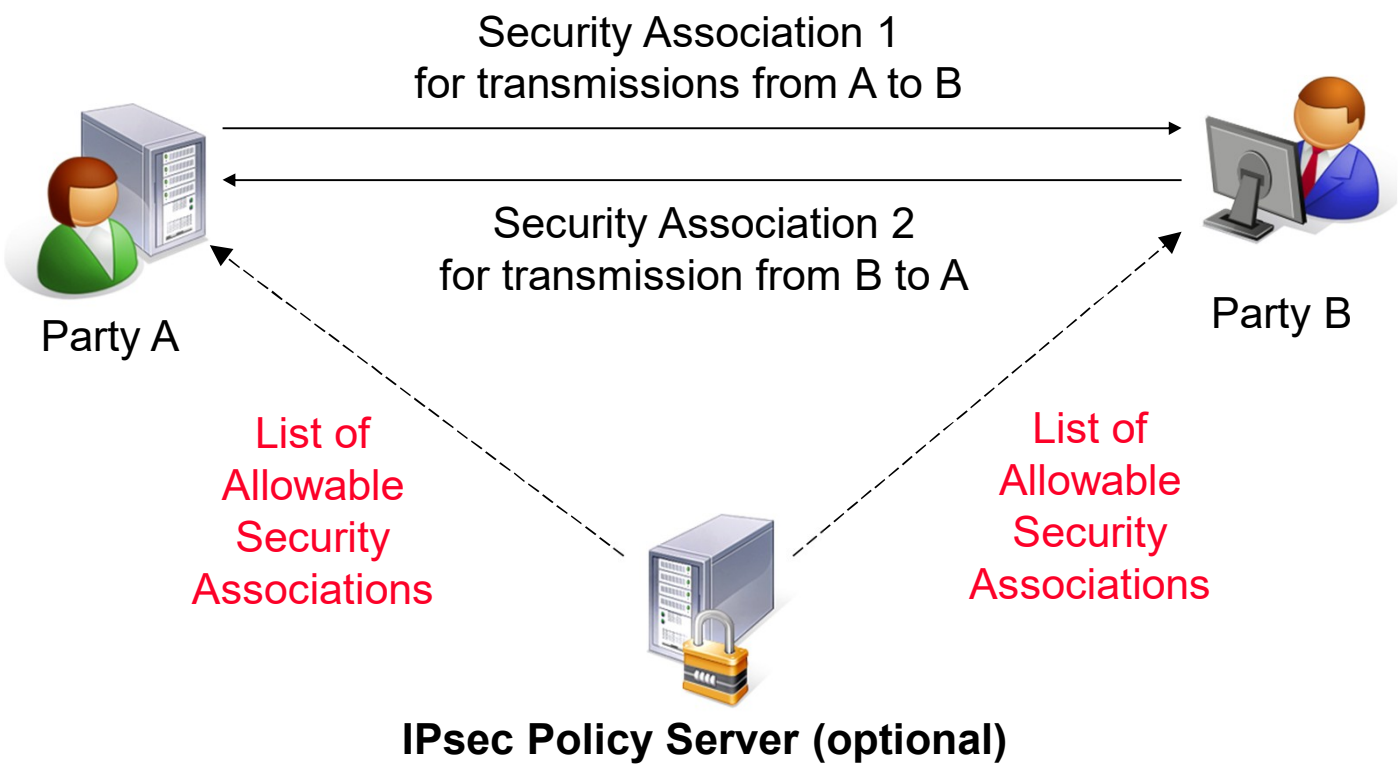
Fig 20.5

Security Associations (SA)

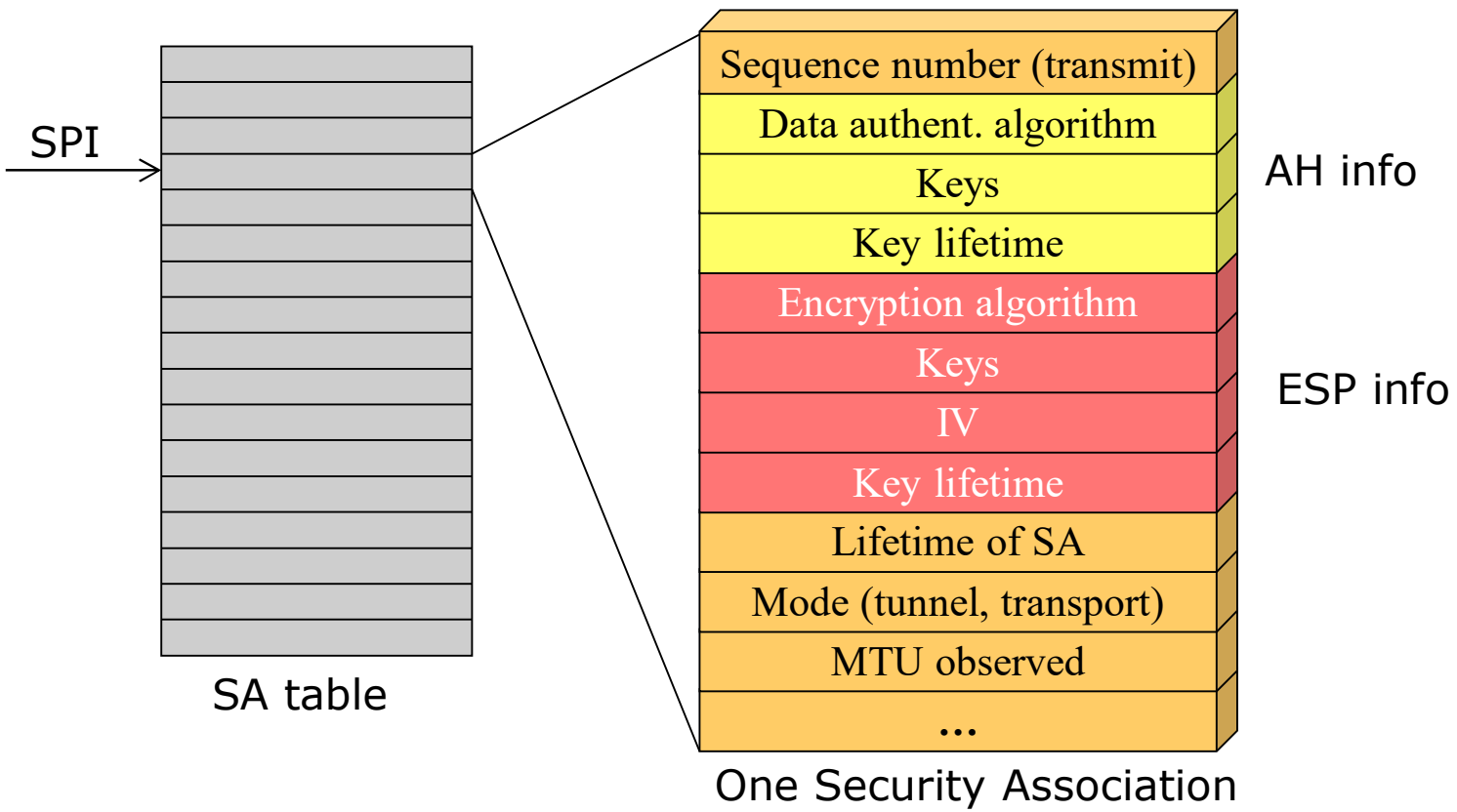


- The SA defines a one-way relationship between sender and receiver
  - Tells how “A” sends traffic to “B”
    (A will offer B a policy. If B accepts this policy, it will send that policy back to A)
  - Two SA:s are required, one for each direction of the communication
  - Can specify: bypass, discard, or how to apply IPsec
- The Security Parameters Index (SPI) tells under what SA a received packet should be processed:
  - SPI is the index used to find the entry for a particular SA
  - Each node has a table with all SA:s
  - SPI is present in each ESP (and AH) header
  - The index is local for two peers (it’s just an index)

Security Associations



SPI, SA table and an SA entry

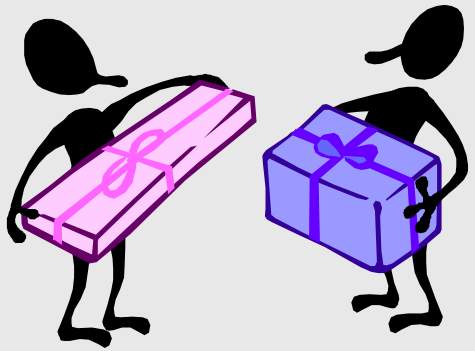


Selectors and the Security Policy Database

- Selectors can be present and determine what SA a particular data packet should belong to:
  - Source and destination IP address
  - Protocol, TOS, port numbers
  - User ID (from OS)
  - Data sensitivity level
  - ...

- Selectors in the Security Policy Database (SPD) in a sender:

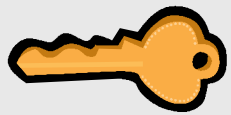
```
src 192.168.0.0/16,  dest 10.0.1.3     port 443, pass          # SSL/TLS traffic
src 192.168.0.0/16,  dest 10.0.0.0/8   port 139, discard       # Windows file sharing
src 192.168.0.0/16,  dest 10.0.0.0/8   port 80,  IPsec: SPI=4  # http
src 192.168.0.0/16,  dest 10.0.0.0/8,            IPsec: SPI=5  # all other traffic to "the 10 network"
```



IKE – Internet Key Exchange

RFC 7296 (140 pages)

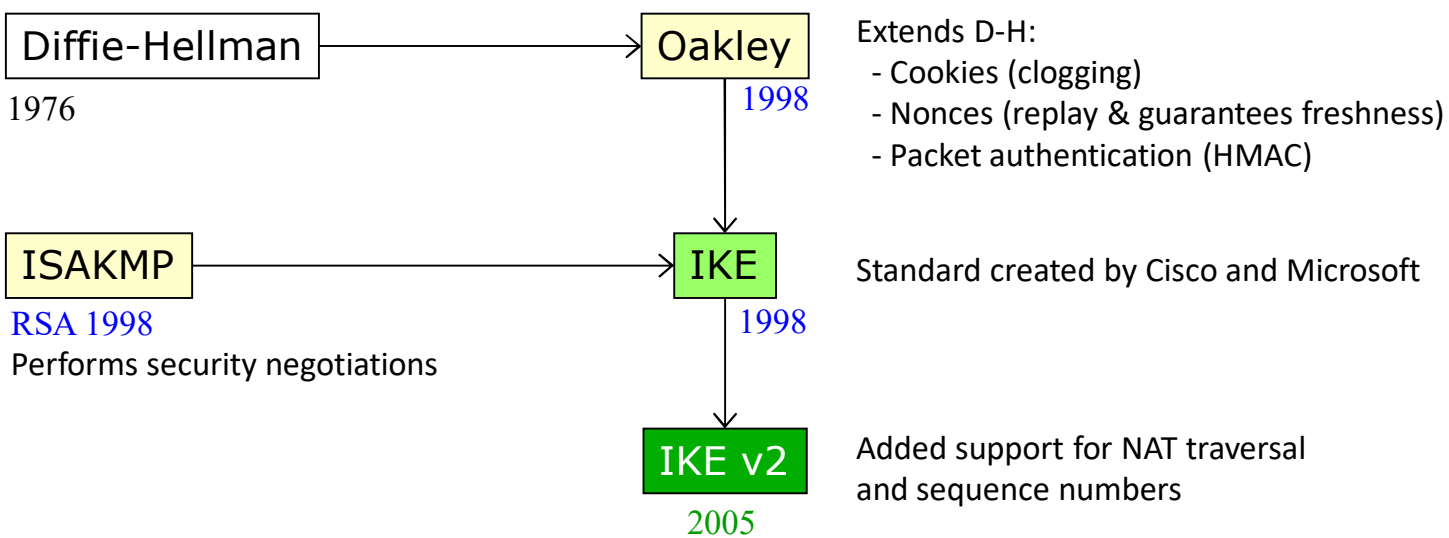
Key management - IKE



- Key management in IPsec
  - Manual configuration of addresses, algorithms and pre-shared keys
  - Automated key management using IKE
- IKE (Internet Key Exchange) is mandatory in IPsec
  - Must be supported by all implementations
- Other protocols "may be used":
  - Certificate-based IKE
  - Kerberos
  - DNSsec - keys from DNS server
  - SKIP (simple key integrity protocol)
  - …
- IKE runs as an application in the system
  - ESP and AH are parts of the system's IP stack

- IKE establishes security associations:
  - Two-way authentication of peers
  - Negotiates security algorithms for the protocols
  - Handles exchange of session crypto keys
- Supports Diffie-Hellman key negotiation
  - Both ephemeral (plain D-H) and pre-defined groups (see SSH lecture)
- IKE v2 based on two protocols:
  - ISAKMP – *Internet Security Association Key Management Protocol*
    - Performs algorithm negotiation
    - Uses UDP port 500
  - Oakley
    - Performs key exchange
    - Uses and extends Diffie-Hellman with cookies, nonces and authentication

IKE overview



IKE (Oakley) extends Diffie-Hellman

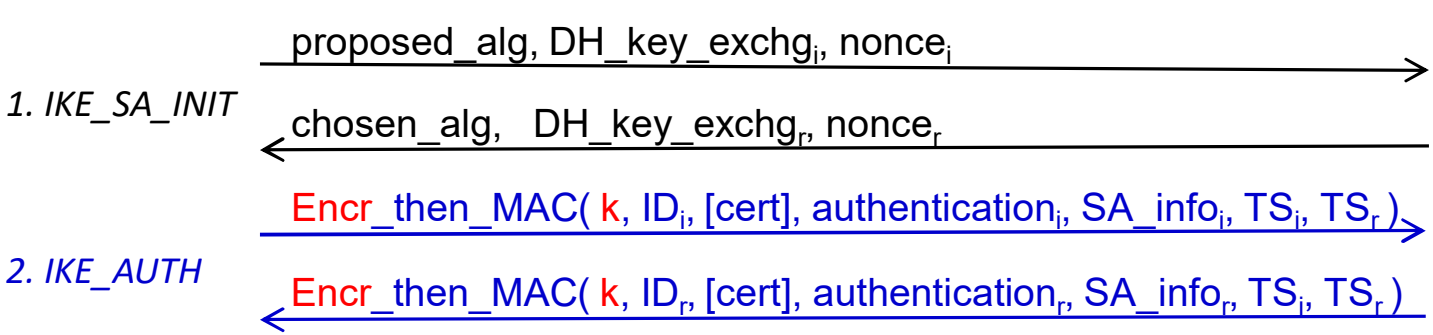
- D-H used for key generation
- Cookies protect against clogging attacks
  - D-H is computational expensive
  - A cookie is sent back to requestor:
    hash(source and destination IP addresses, port numbers, secret)
  - Must be returned before computation starts
  - Server does not have to store any state of this request
  - It should not be possible to reuse a cookie by an attacker
  - If cookie comes back, IP address is ok

- nonces protect against replay attacks (guarantees freshness)
- HMAC packet authentication against MITM attacks
- Supports Perfect Forward Secrecy, PFS

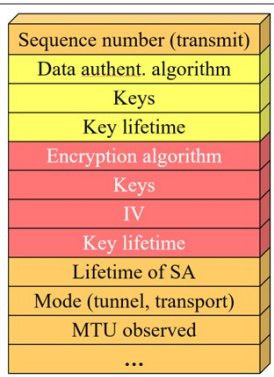
IKEv2 Protocol [RFC 7296]

Two rounds/phases: (somewhat simplified)

1. Negotiate algorithms and exchange D-H parameters
2. Authentication, create SAs and derive keys for IPsec



Authentication of parties and of the earlier messages. Result is an SA with keys:



i=initiator, r=respondent
k=key from DH or pre-shared key
SA_info = algorithms and parameters for IPsec SA creation
TS = Traffic selectors: IP address range, port range and protocol for the tunnel (from SPD database) on the initiator's and respondent's side. Fourth message is a confirmation of TSi and TSr.

Fig 20.11 https://study-ccnp.com/internet-key-exchange-ikev1-ikev2-overview

Key generation and Traffic selectors

- The use of D-H guarantees Perfect Forward Secrecy, PFS
  - Creation of master secret: SKEYSEED = prf ($nonce_i$ || $nonce_r$, DH)
  - Key generation: prf (SKEYSEED, $nonce_i$ || $nonce_r$ || SPI_i || SPI_r)

- Generated keys are for each direction:
  SK_e for encryption
  SK_a for autentication (integrity)

```
SK_e  i:7ee71f3b1168b19b656e39575e985466fa86a71f802d55e6
      r:2e43283551a2408a1b8ebf16769d748118e439f2591ab562
SK_a  i:ab331c5718cc21811e8bd35313a17c6149d0a7f4
      r:6111429868ff314520d43c12523b23f06e6f9e7d
```

IKE also needs some keys for internal use (for rekeying, etc)

- Traffic selectors:

```
traffic selectors (i):
  0 type 7 protocol_id 0 addr 192.168.3.0 - 192.168.3.255 port 0 - 65535
traffic selectors (r):
  0 type 7 protocol_id 0 addr 192.168.5.0 - 192.168.5.255 port 0 - 65535
```

IPsec and IKE fingerprinting

- Possible to detect hosts/devices supporting IPsec (i.e. IKE):

```
% ipsecscan 10.0.0.1 10.0.0.10
10.0.0.5 IPSec status: Enabled
```

- A negative reply does not mean that the host does not support or use IPsec
- IKE replies can be used to fingerprint the system (UDP port 500):

```
root@kali:~/ike-scan# ike-scan [redacted] -M [redacted]
Starting ike-scan 1.9.4 with 1 hosts (http://www.nta-monitor.com/tools/ike-scan/)
[redacted]      Main Mode Handshake returned
        HDR=(CKY-R=1f9e7509cf33c00f)
        SA=(Enc=3DES Hash=MD5 Group=2:modp1024 Auth=PSK LifeType=Seconds LifeDuration=28800)

IKE Backoff Patterns:  (Backoff = retransmission)

IP Address      No.     Recv time               Delta Time
[redacted]      1       1456756249.384123       0.000000
[redacted]      Implementation guess: Linksys Etherfast

Ending ike-scan 1.9.4: 1 hosts scanned in 60.452 seconds (0.02 hosts/sec).  1 returned handshake; 0 returned r
```

- IKE-scan can figure out the vendor of the system
- Knowing type of system, specific vulnerabilities may exist
  - IKE packet payload buffer overflow (Cisco, CheckPoint)
  - IKE response buffer overflow (multiple vendors)
  - ISAKMP packet DoS (Cisco)

IPsec and NAT/PAT

- IKE v2 supports NAT/PAT
- AH is not compatible in transport mode
  - HMAC covers original source IP header which is modified by NAT
- ESP is not compatible with NAT/NAPT
  - NAT gateway must modify TCP and UDP checksum when IP address changes – not possible
  - NAT (NAPT) also modifies port numbers, but ESP has no ports
- Solution called NAT-Traversal, NAT-T [RFC 3948]
  - IKE can detect if NAT devices are present in the transmission path
  - If so, it tunnels ESP packets in a UDP connection instead – port 4500
  - Original IP address (NAT-OA) also included to allow receiving system to verify it (IKE)
  - Windows requires a registry change to allow NAT-T
- Not uncommon that border firewalls block port 500 and 4500
  - "Firewall friendlier" solution is to use TLS-based solutions (e.g. tunnel traffic with TLS and using port 443) even if it is not web traffic (Good and bad…)

Microsoft

0 = the default value: Windows can't establish security associations with servers located behind NAT devices.

1 = Windows can establish security associations with servers that are located behind NAT devices.

2 = Windows can establish security associations when both the server and VPN client are behind NAT devices.

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\IPSec\AssumeUDPEncapsulationContextOnSendRule

See RFC 3715 for more details

Summary IPsec

- Two modes: Tunnel mode and Transport mode
- Three protocols: AH, ESP, IKE for key negotiation
- SA, SPI, Selectors
  - Each packet contains an SPI – an index into the SA
  - Selectors determine what SA an outgoing packet belongs to [WHAT]
  - SA is an array with connection-related info (keys, etc.) [HOW]