Network Level Security – IPSec

Vivek Kumar Anand

IP Security

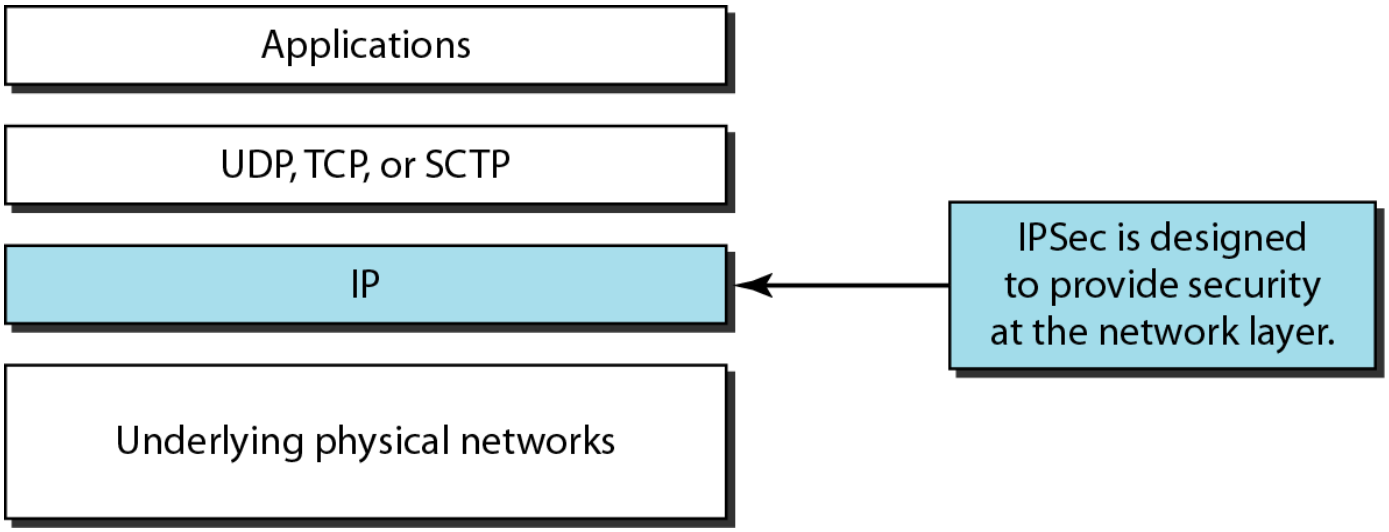
IP Security

- have a range of application specific security mechanisms
  - eg. S/MIME, PGP, Kerberos, SSL/HTTPS
- however there are security concerns that cut across protocol layers
- would like security implemented by the network for all applications

IP Security

- general IP Security mechanisms provides
  - authentication
  - confidentiality
  - key management
- applicable to use over LANs, across public & private WANs, & for the Internet
- need identified in 1994 report
  - need authentication, encryption in IPv4 & IPv6

IPSecurity (IPSec) is a collection of protocols designed by the Internet Engineering Task Force (IETF) to provide security for a packet at the network level.

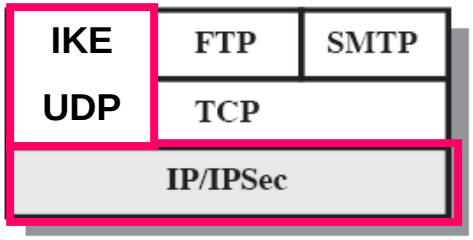


Outline

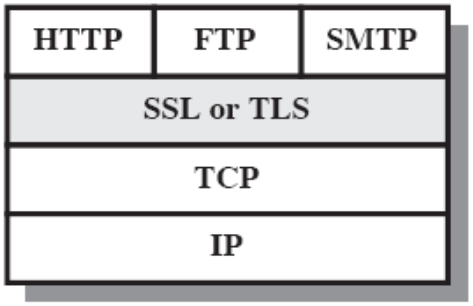
IPSec

- Modes and Protocols
- IKE Protocol Basics

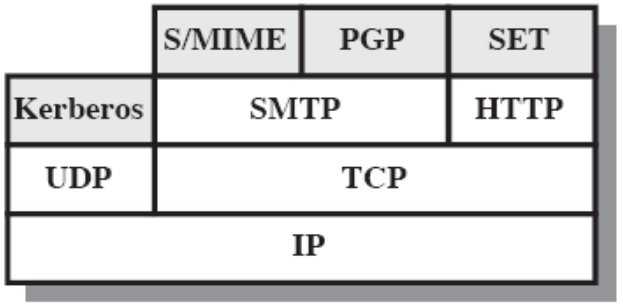
Which Layer to Add Security to?



(a) Network Level



(b) Transport Level



(c) Application Level

Relative Location of Security Facilities in the TCP/IP Protocol Stack

Why IP Security?

- IP datagrams have no inherent security
  - IP source address can be spoofed
  - Content of IP datagrams can be sniffed
  - Content of IP datagrams can be modified
  - IP datagrams can be replayed
- IPSec is a method for protecting IP datagrams
  - Standardized by IETF: dozens of RFCs.
  - Only sender and receiver have to be IPsec compliant
    - Rest of network can be regular IP

What is security at the network-layer?

Between two network entities:

- Sending entity encrypts/authenticates the payloads of datagrams. Payload could be:
  - TCP segment, UDP segment, ICMP message, OSPF (routing) message, and so on.
- All data sent from one entity to the other would be hidden/authenticated:
  - Web pages, e-mail, P2P file transfers, TCP SYN packets, and so on.
- That is, “blanket coverage”.

IPsec

- IPSec provides
  - Access control: User authentication
  - Data integrity
  - Data origin authentication
  - Rejection of replayed packets
  - Confidentiality (encryption)
  - Limited traffic flow confidentiality
- Benefits:
  - Security at Layer 3 ⇒ Applies to all transports/applications
  - Can be implemented in Firewall/router ⇒ Security to all traffic crossing the perimeter
  - Transparent to applications and can be transparent to end users
  - Can provide security for individual users
- Applications: VPNs, Branch Offices, Remote Users, Extranets

Virtual Private Networks (VPNs)

- Institutions often want private networks for security.
  - Costly! Separate routers, links, DNS infrastructure.
- With a VPN, institution's inter-office traffic is sent over public Internet instead.
  - But inter-office traffic is encrypted and/or authenticated before entering public Internet

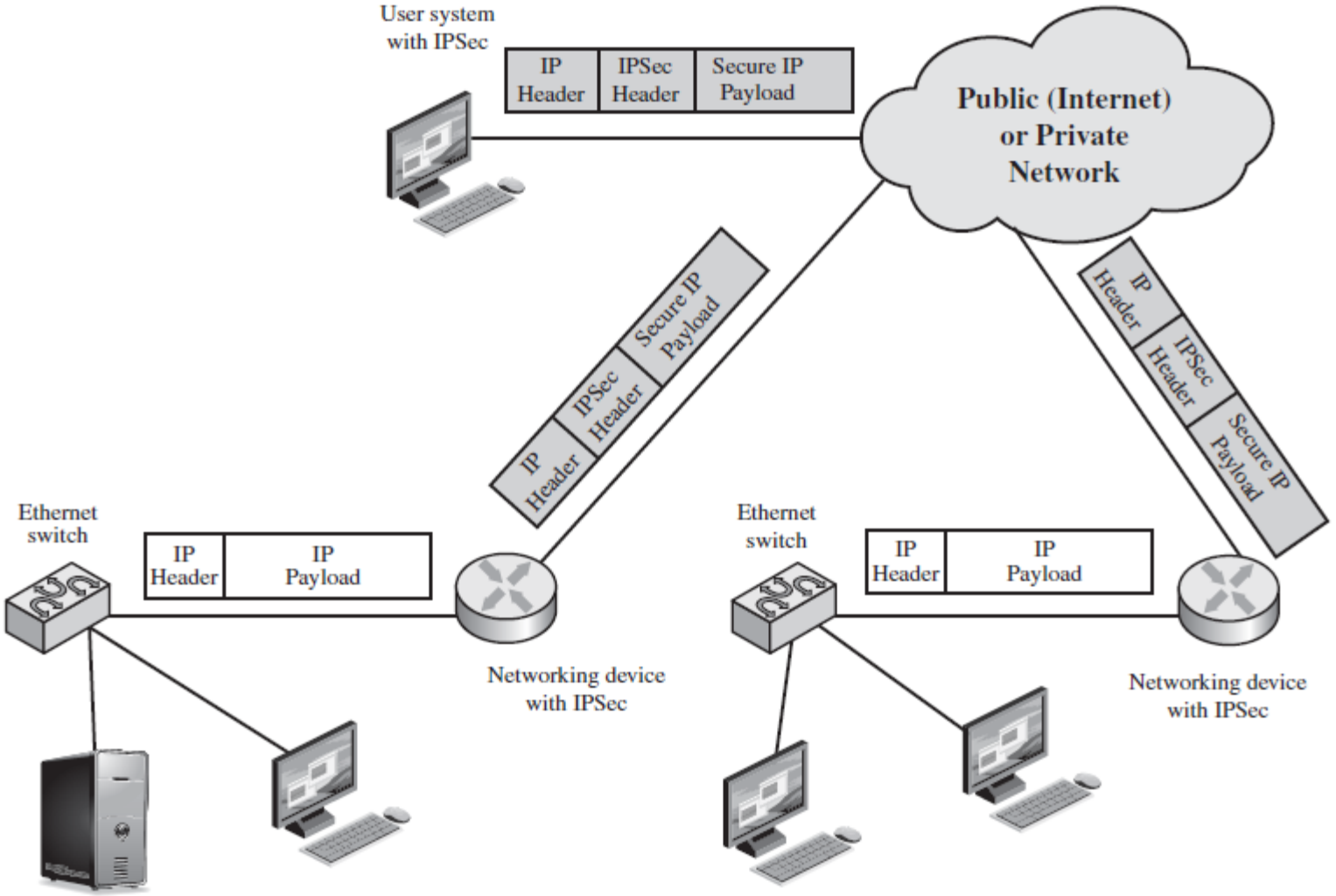
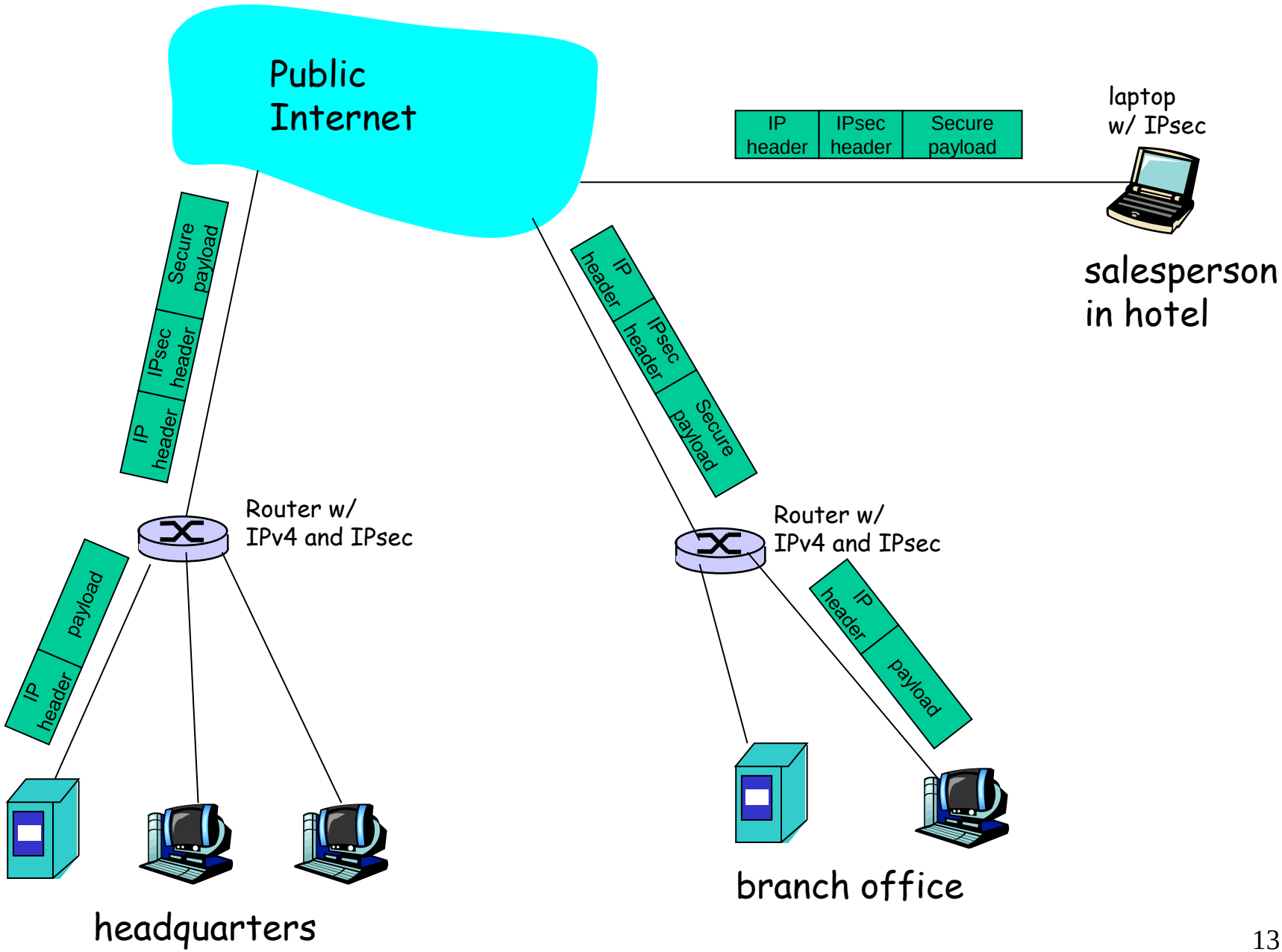


Figure 20.1 An IP Security Scenario

Virtual Private Network (VPN)



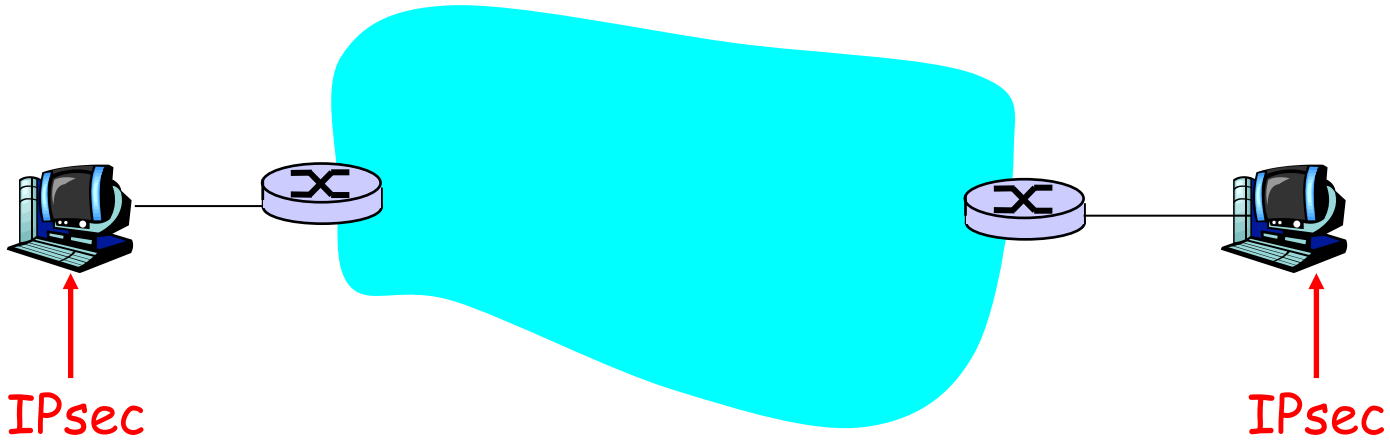
IPsec

- Two modes:
  - Transport and Tunnel
- Two protocols providing different service models:
  - AH – Authentication Header
  - ESP – Encrypted Security Payload

IPSec Modes: Transport and Tunnel Modes

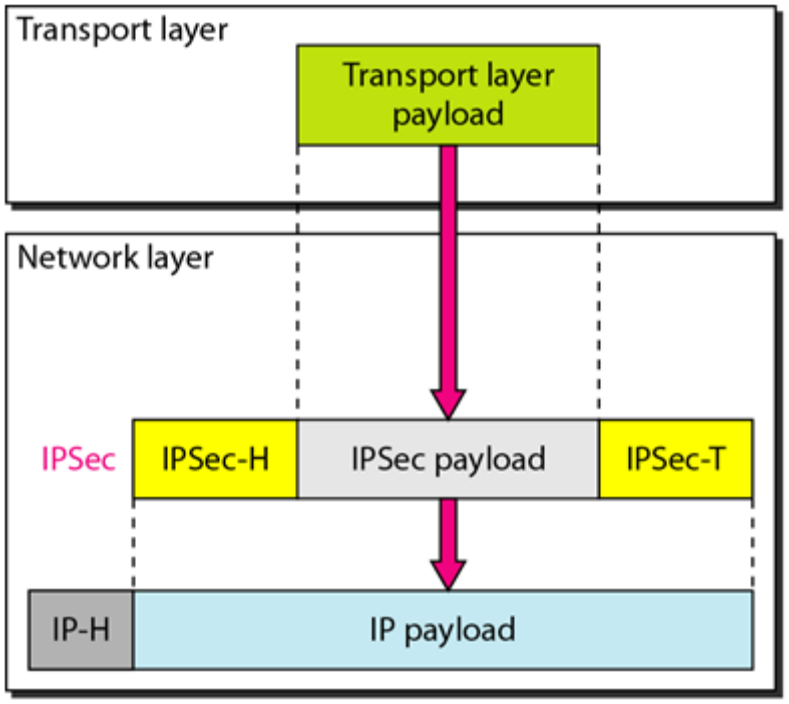
- Transport Mode provides protection primarily for upper-layer protocols – that is, for the IP datagram payload.
  - Transport mode is typically used in end-to-end communication between two hosts.
- Tunnel Mode extends protection to the entire datagram, by encapsulating it in a new "outer" datagram.
  - Tunnel mode is typically used in communication between two routers (must be used if a router is involved).

IPsec Transport Mode



- IPsec datagram emitted and received by end-systems.
- Protects upper level protocols

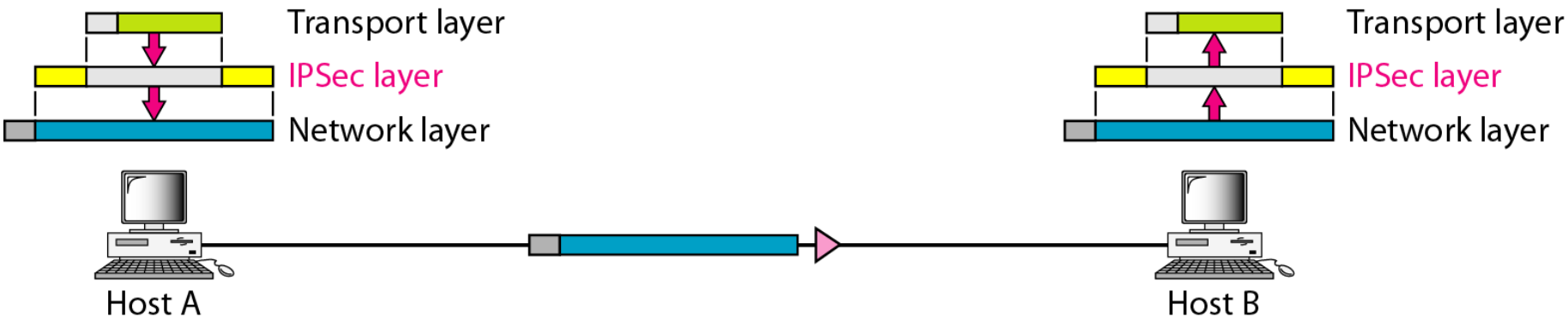
Figure *Transport mode and tunnel modes of IPSec protocol*



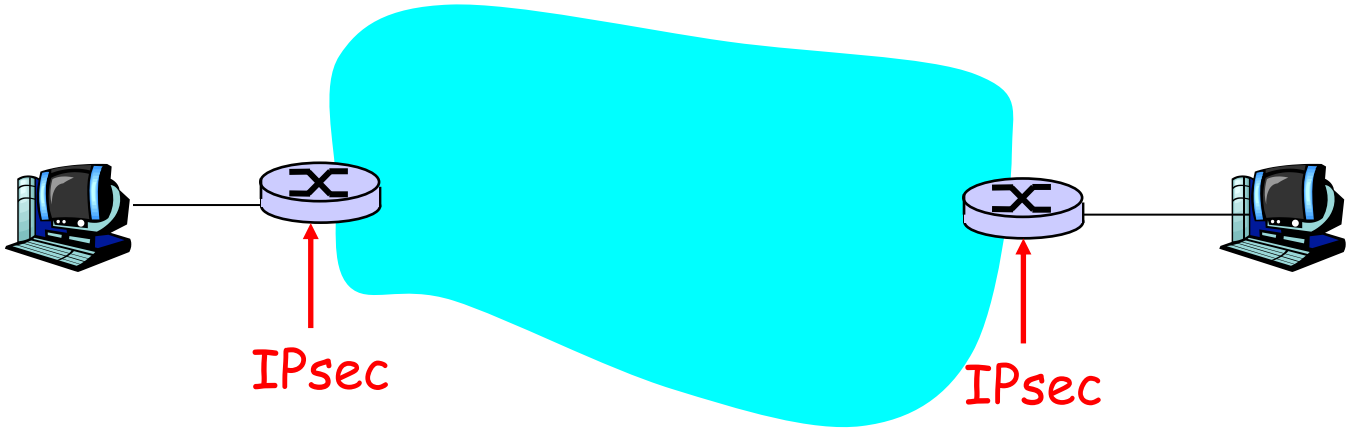
a. Transport mode

IPSec in the transport mode does not protect the IP header; it only protects the information coming from the transport layer.

Figure *Transport mode in action*

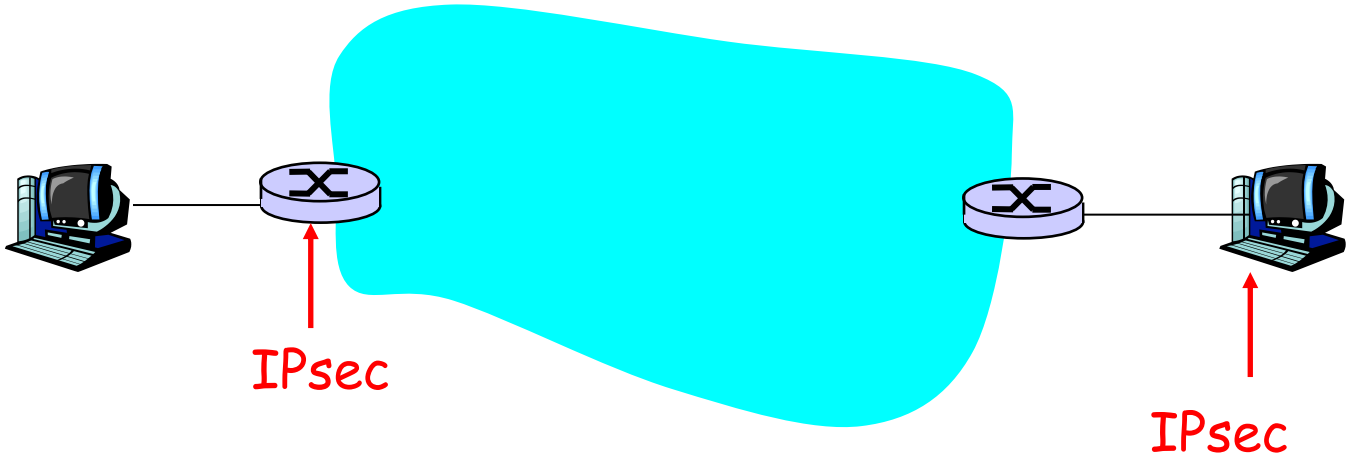


IPsec – tunneling mode (1)



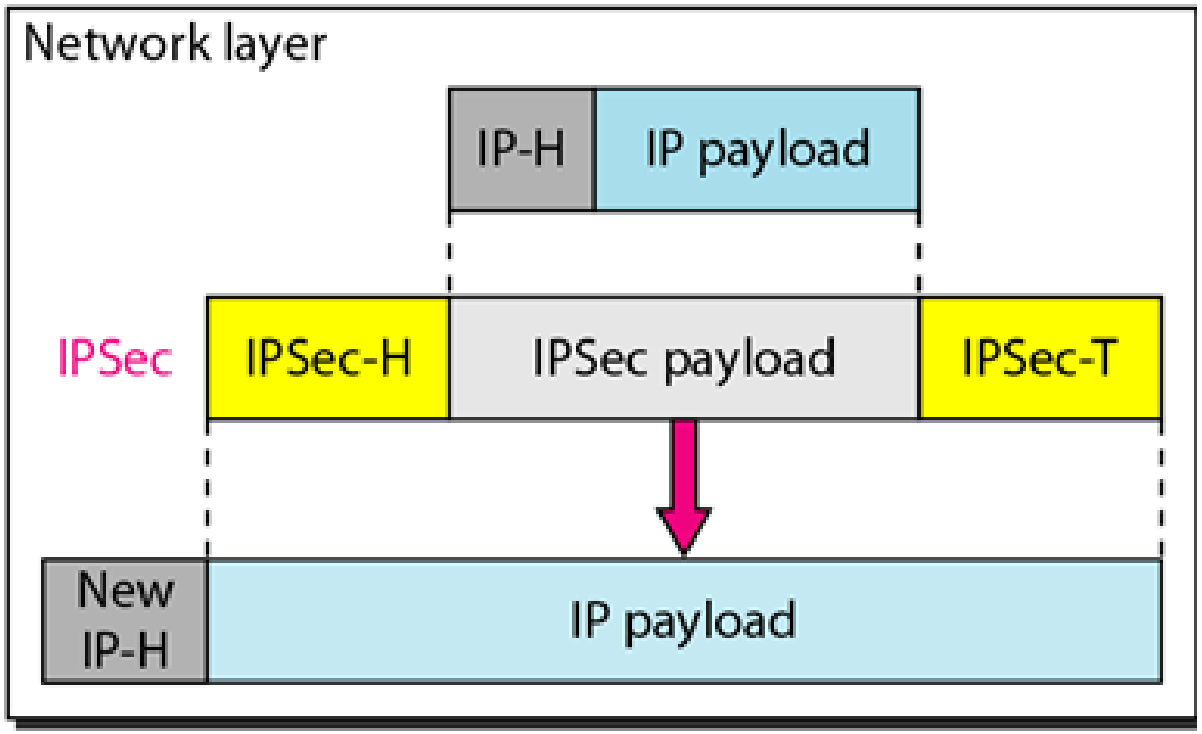
- End routers are IPsec aware. Hosts need not be.

IPsec – tunneling mode (2)



- Also tunneling mode.

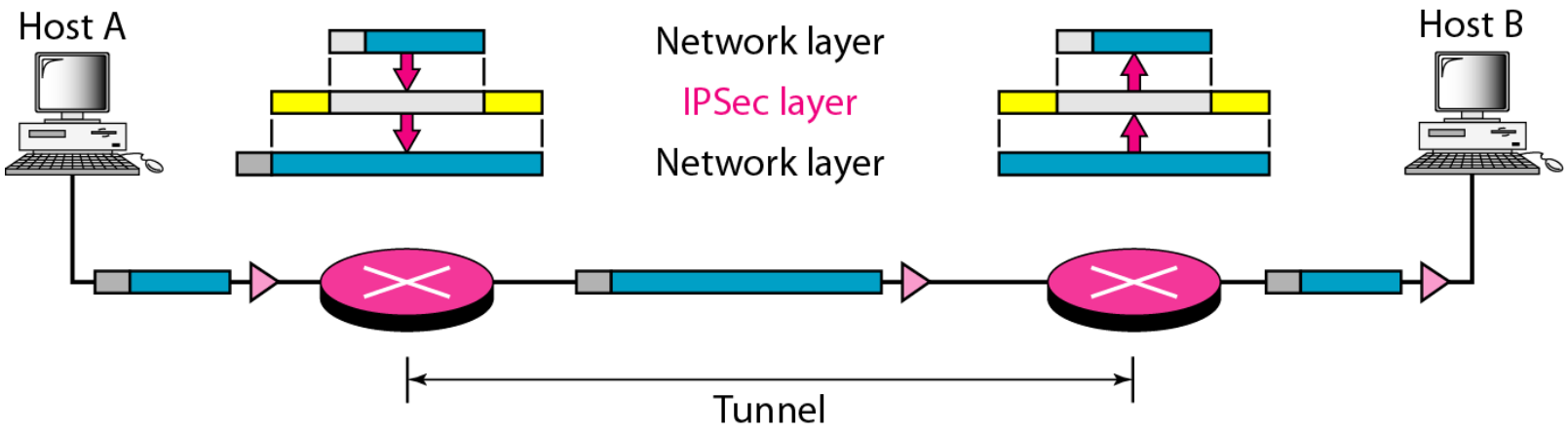
IPsec – tunneling mode (2)



b. Tunnel mode

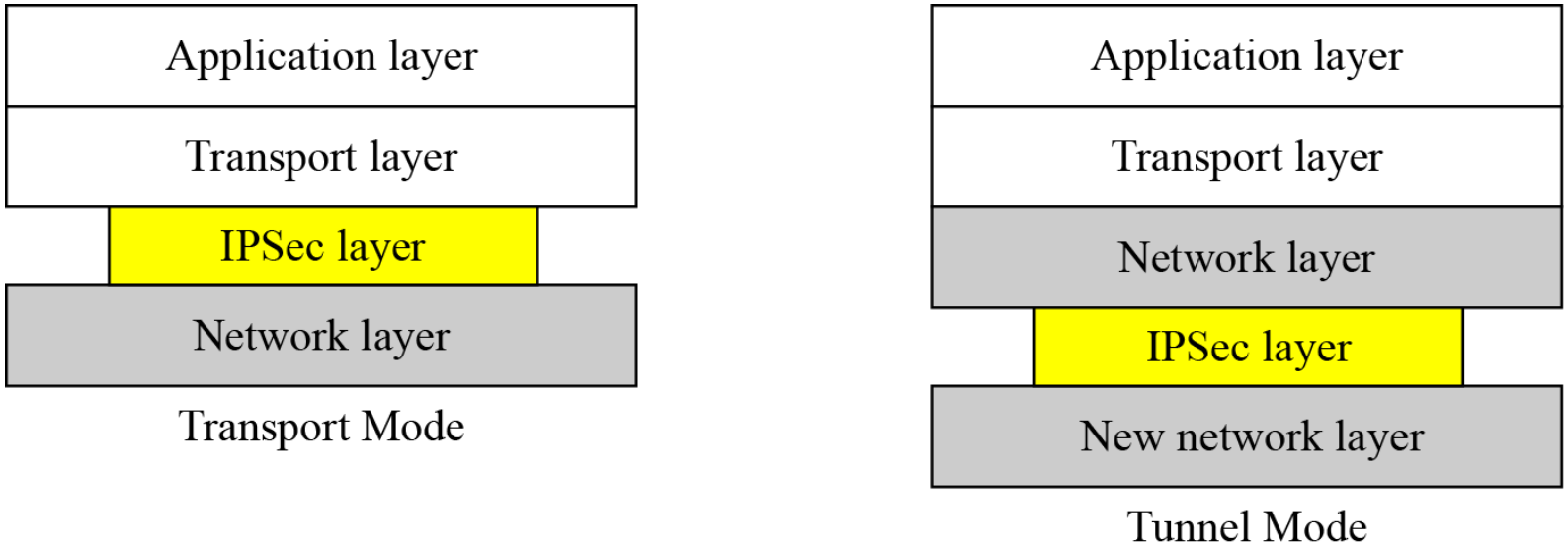
IPSec in tunnel mode protects the original IP header.

Figure *Tunnel mode in action*



Comparison

Figure *Transport mode versus tunnel mode*



TWO SECURITY PROTOCOL

IPSec defines two protocols—the Authentication Header (AH) Protocol and the Encapsulating Security Payload (ESP) Protocol—to provide authentication and/or encryption for packets at the IP level.

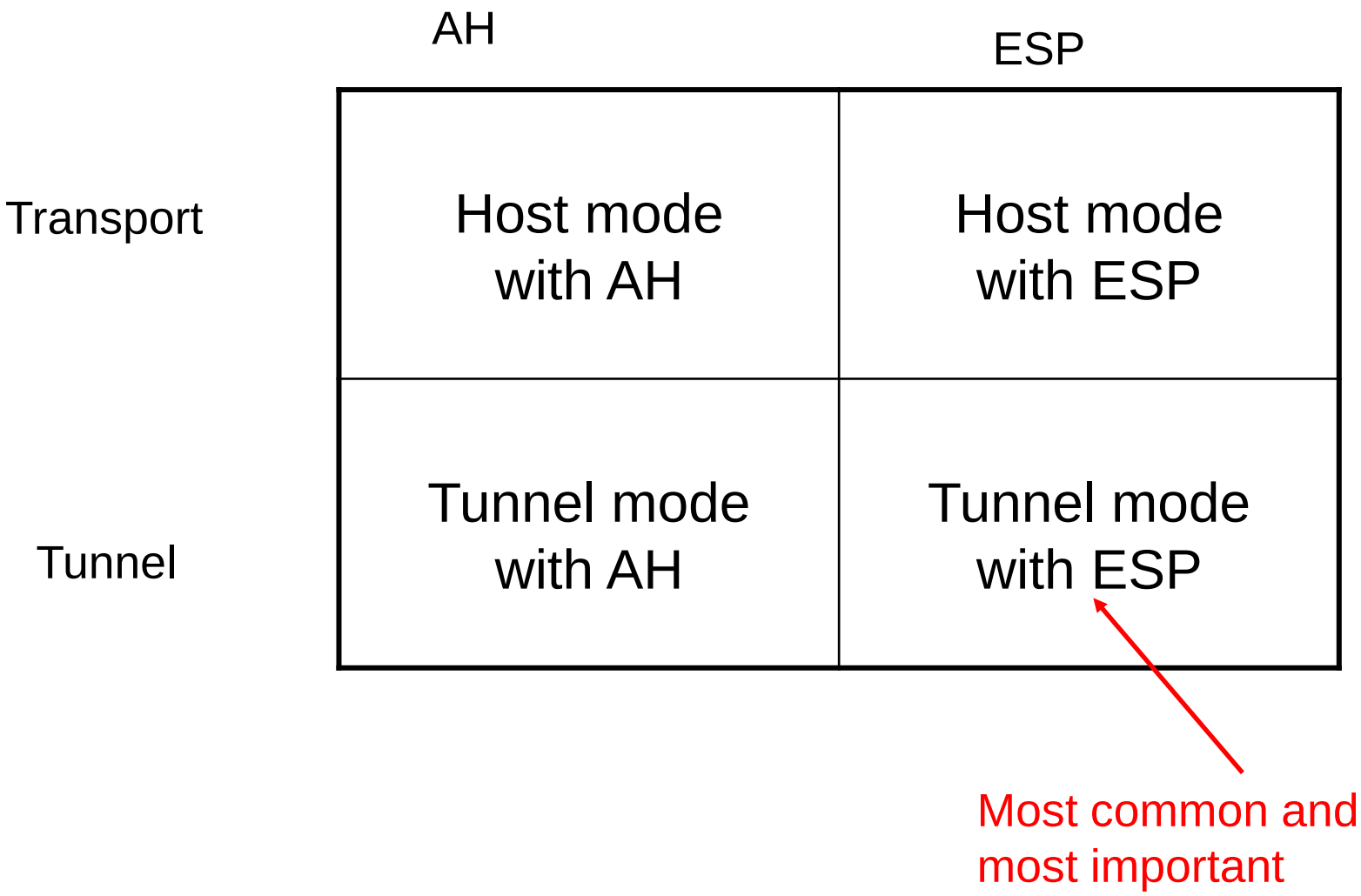
Topics

Authentication Header (AH)
Encapsulating Security Payload (ESP)
IPv4 and IPv6
AH versus ESP
Services Provided by IPSec

IPSec protocols

- Authentication Header (AH) protocol
  - provides source authentication & integrity but *not* confidentiality
- Encapsulation Security Protocol (ESP)
  - provides source authentication, integrity, *and confidentiality*
  - more widely used than AH

Four combinations are possible!



IP Security Architecture

- Internet Key Exchange (IKE)
- IPSec
- Security Association Database
- Security Policy database

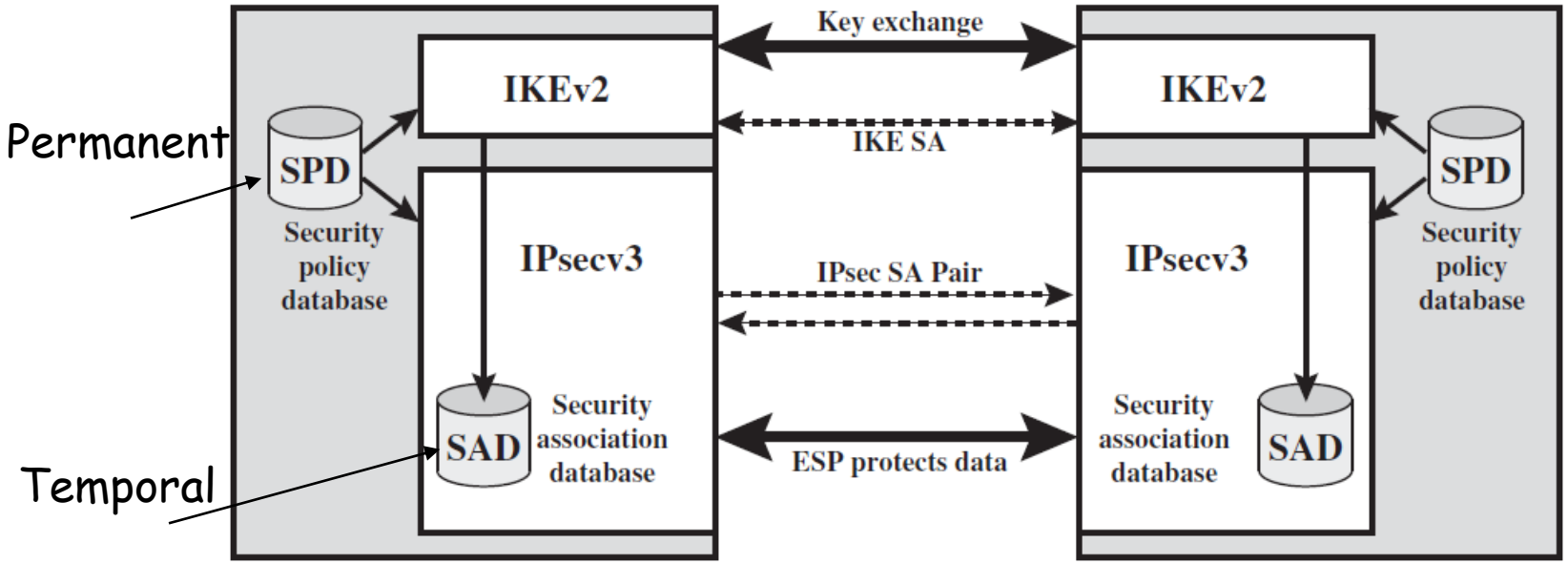


Figure 20.2 IPsec Architecture

IPsec Architecture

Security Association Database

- Each host has a database of Security Associations (SAs) (secure connection)
- SA = One-way security relationship between sender & receiver Two- way may use different security ⇒Two SA's required
- Defined by 3 parameters: ➢Security Parameters Index (SPI)
  - ➢IP Destination Address
  - ➢Security Protocol Identifier: AH or ESP

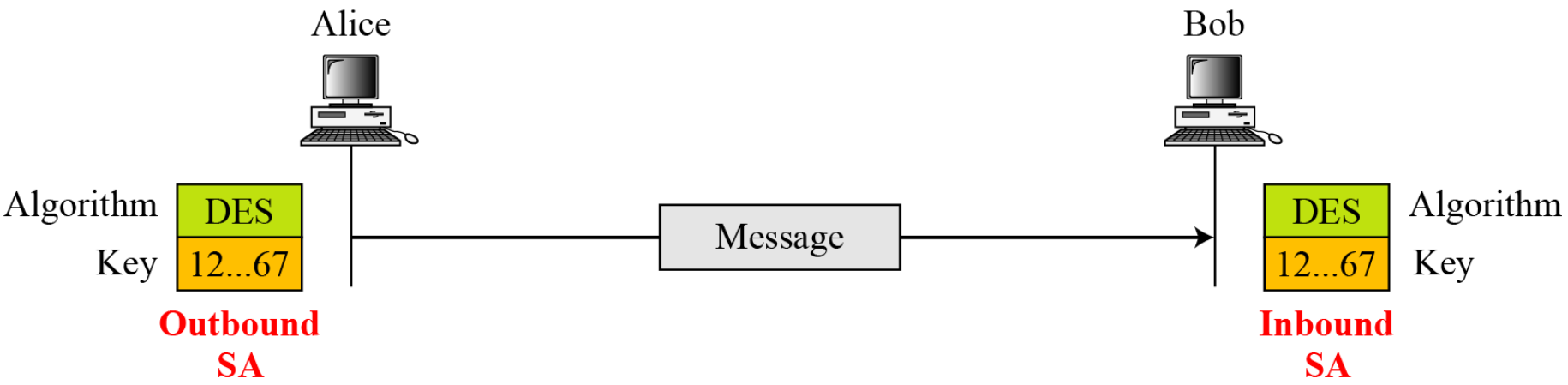
- For each SA, the database contains: ➢SPI
  - ➢Sequence number counter and counter overflow flag
  - ➢Anti-replay window
  - ➢AH Information and ESP information
  - ➢Lifetime of the SA
  - ➢Mode: Transport or tunnel or wildcard
  - ➢Path MTU

Security Association

Security Association is a very important aspect of IPSec. IPSec requires a logical relationship, called a Security Association (SA), between two hosts.

Idea of Security Association

Figure *Simple SA*



Security Association Database (SAD)

Figure *SAD*

Index	SN	OF	ARW	AH/ESP	LT	Mode	MTU
< SPI, DA, P >							
< SPI, DA, P >							
< SPI, DA, P >							
< SPI, DA, P >							

Security Association Database

Legend:

SPI: Security Parameter Index

DA: Destination Address

AH/ESP: Information for either one

P: Protocol

Mode: IPSec Mode Flag

SN: Sequence Number

OF: Overflow Flag

ARW: Anti-Replay Window

LT: Lifetime

MTU: Path MTU (Maximum Transfer Unit)

(Continued)

Table *Typical SA Parameters*

Parameters	*Description*
Sequence Number Counter	This is a 32-bit value that is used to generate sequence numbers for the AH or ESP header.
Sequence Number Overflow	This is a flag that defines a station's options in the event of a sequence number overflow.
Anti-Replay Window	This detects an inbound replayed AH or ESP packet.
AH Information	This section contains information for the AH protocol: 1. Authentication algorithm 2. Keys 3. Key lifetime 4. Other related parameters
ESP Information	This section contains information for the ESP protocol: 1. Encryption algorithm 2. Authentication algorithm 3. Keys 4. Key lifetime 5. Initiator vectors 6. Other related parameters
SA Lifetime	This defines the lifetime for the SA.
IPSec Mode	This defines the mode, transport or tunnel.
Path MTU	This defines the path MTU (fragmentation).

SECURITY POLICY

Another import aspect of IPSec is the Security Policy (SP), which defines the type of security applied to a packet when it is to be sent or when it has arrived. Before using the SAD, a host must determine the predefined policy for the packet.

Topics

Security Policy Database

Security Policy Database

- Relates IP traffic to specific SAs
  - Match subset of IP traffic to relevant SA
  - Use selectors to filter outgoing traffic to map
  - Based on: local & remote IP addresses, next layer protocol, name, local & remote ports

Protocol	Local IP	Port	Remote IP	Port	Action	Comment
UDP	1.2.3.101	500	*	500	BYPASS	IKE
ICMP	1.2.3.101	*	*	*	BYPASS	Error messages
*	1.2.3.101	*	1.2.3.0/24	*	PROTECT: ESP intransport-mode	Encrypt intranet traffic
TCP	1.2.3.101	*	1.2.4.10	80	PROTECT: ESP intransport-mode	Encrypt to server
TCP	1.2.3.101	*	1.2.4.10	443	BYPASS	TLS: avoid double encryption
*	1.2.3.101	*	1.2.4.0/24	*	DISCARD	Others in DMZ
*	1.2.3.101	*	*	*	BYPASS	Internet

(Continued) Connection identifiers

Each entry in SPD can be accessed using tuple
<SA,DA,Name,Proto,Sport,Dport> (it works as index)
SA and DA can be uni,multicast or wildcard addresses
Name usually is a DNS entry, Protocolis either AH or ESP
Port is the process port

Index	Policy
< SA, DA, Name, P, SPort, DPort >	
< SA, DA, Name, P, SPort, DPort >	
< SA, DA, Name, P, SPort, DPort >	
< SA, DA, Name, P, SPort, DPort >	

Legend:

SA: Source Address
DA: Destination Address
P: Protocol
SPort: Source Port
DPort: Destination Port

(Continued)

When a packet is to be sent,the outbound SPD is consulted.

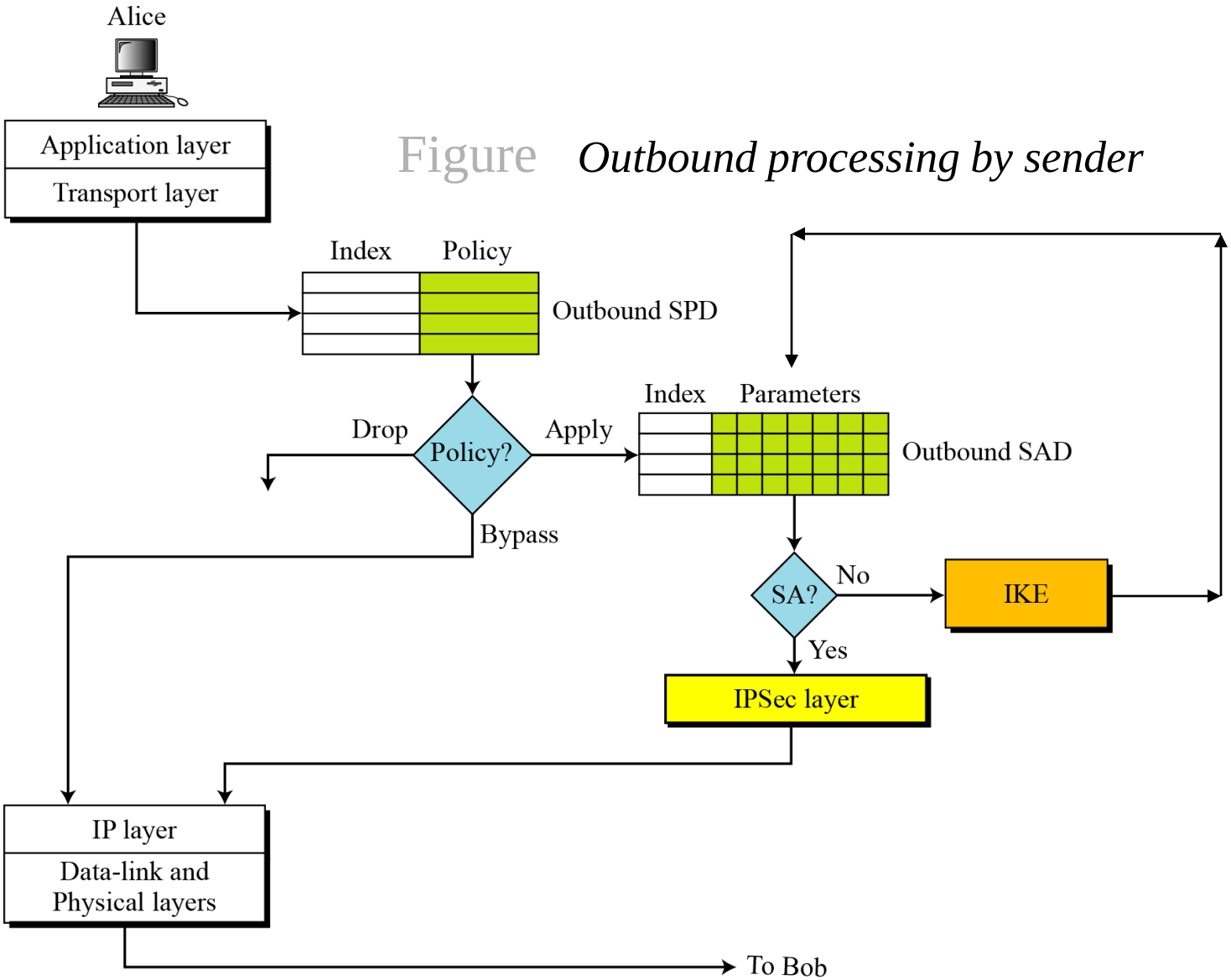
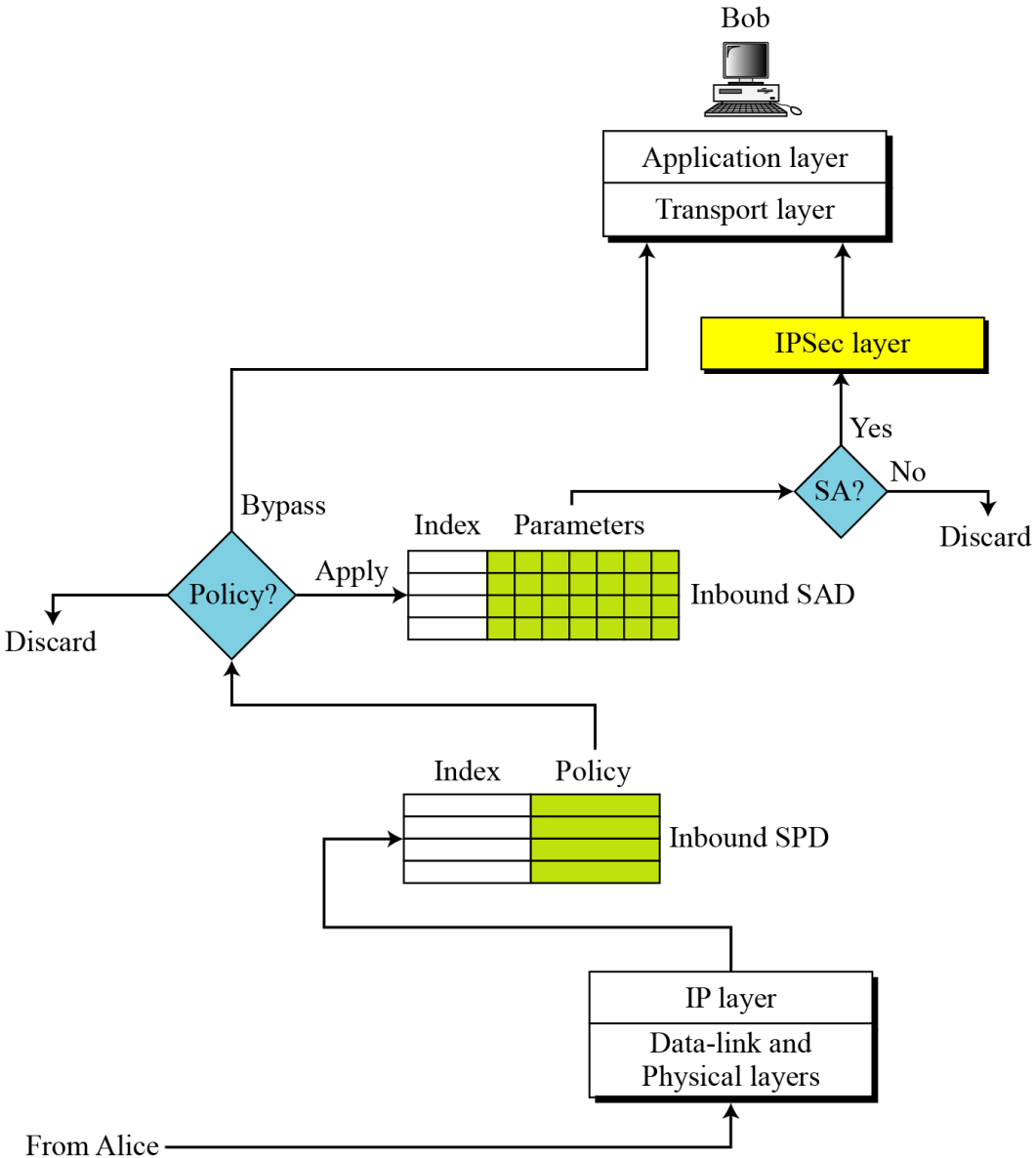


Figure *Outbound processing by sender*

(Continued)

Figure *Inbound processing*



INTERNET KEY EXCHANGE (IKE)

The Internet Key Exchange (IKE) is a protocol designed to create both inbound and outbound Security Associations.

Topics

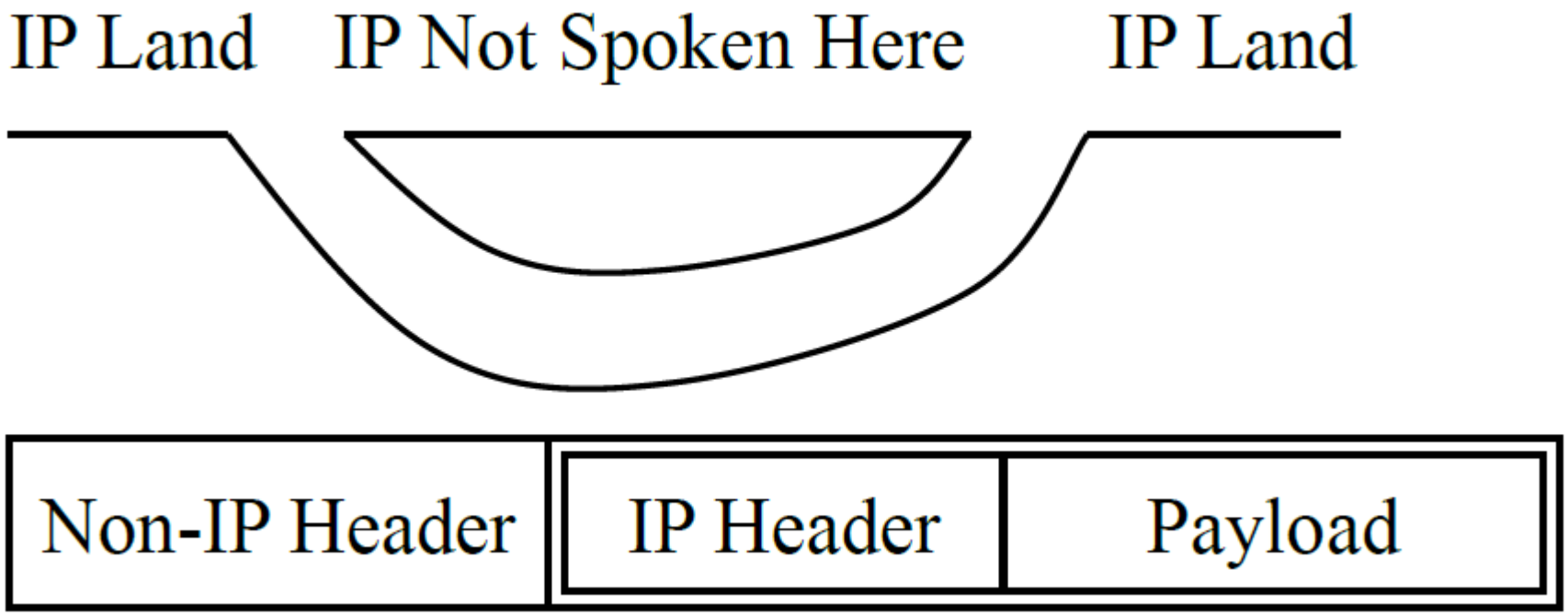
Improved Diffie-Hellman Key Exchange
IKE Phases
Phases and Modes
Phase I: Main Mode
Phase I: Aggressive Mode
Phase II: Quick Mode
SA Algorithms

Security Associations (SAs)

- Before sending data, a virtual connection is established from sending entity to receiving entity.
- Called "security association (SA)"
  - SAs are simplex: for only one direction
- Both sending and receiving entities maintain *state information* about the SA
  - Recall that TCP endpoints also maintain state information.
  - IP is connectionless; IPsec is connection-oriented!
- How many SAs in VPN w/ headquarters, branch office, and n traveling salesperson?

Tunnel

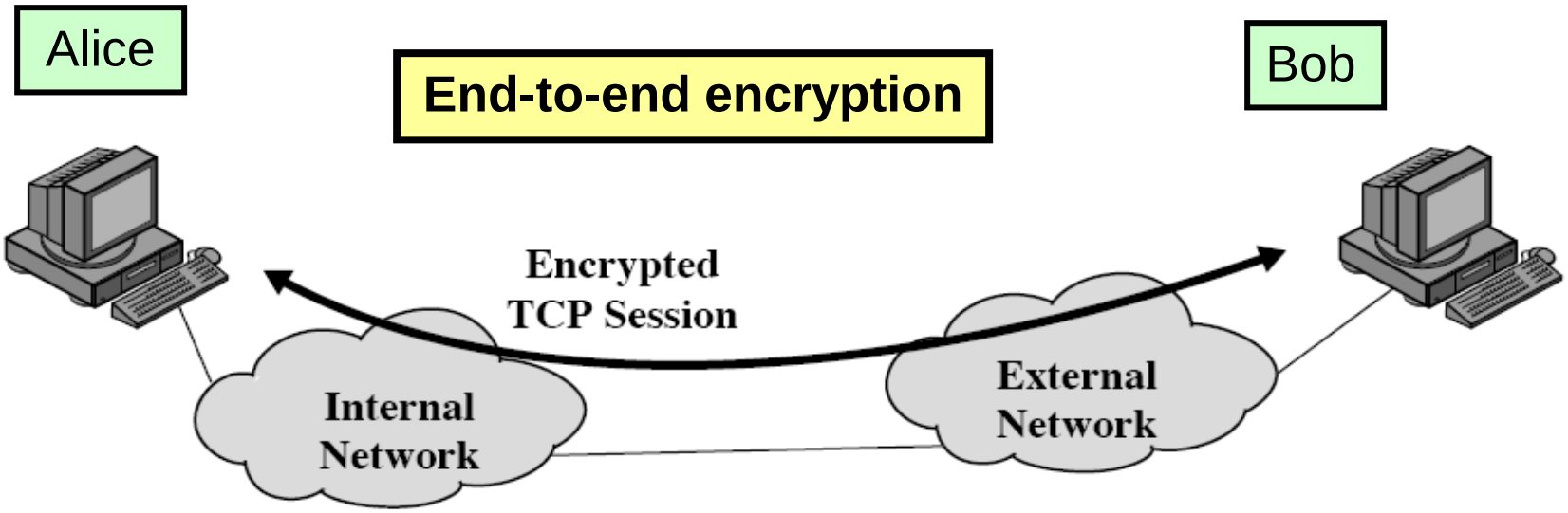
- Tunnel = Encaptulation
- Used whenever some feature is not supported in some part of the network, e.g., multicasting, mobile IP



Tunnel Mode and Transport Mode Functionality

	Transport Mode SA	Tunnel Mode SA
AH	Authenticates IP payload and selected portions of IP header and IPv6 extension headers.	Authenticates entire inner IP packet (inner header plus IP payload) plus selected portions of outer IP header and outer IPv6 extension headers.
ESP	Encrypts IP payload and any IPv6 extension headers following the ESP header.	Encrypts entire inner IP packet.
ESP with Authentication	Encrypts IP payload and any IPv6 extension headers following the ESP header. Authenticates IP payload but not IP header.	Encrypts entire inner IP packet. Authenticates inner IP packet.

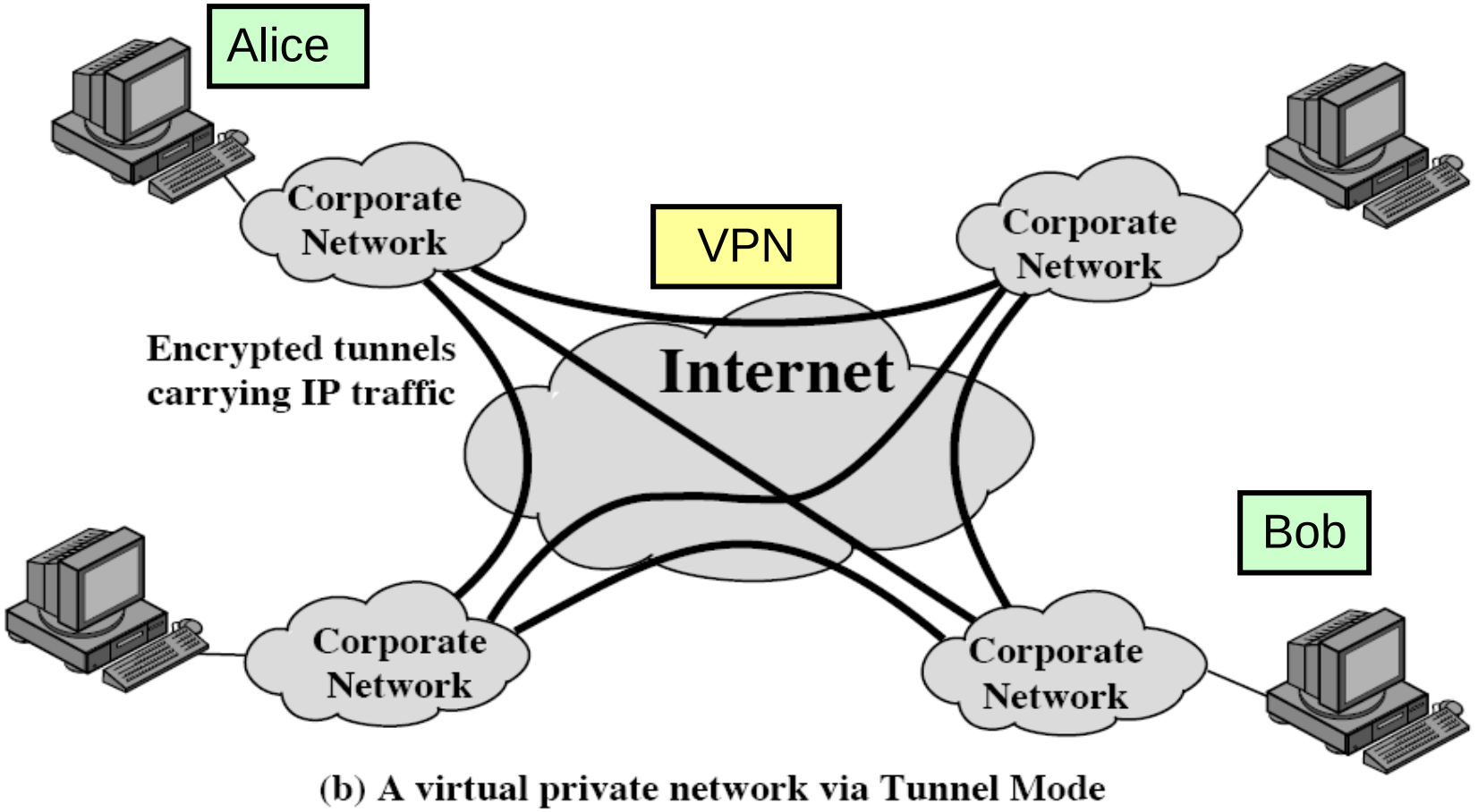
ESP in transport mode



(a) Transport-level security

ESP in transport mode conceals what Alice is saying to Bob, but not that Alice and Bob are communicating.

ESP in tunnel mode

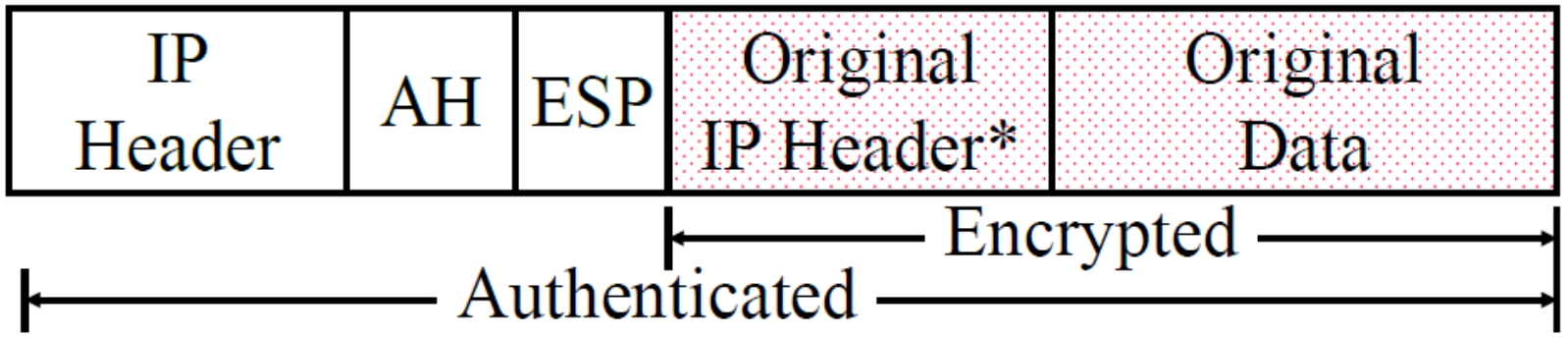


(b) A virtual private network via Tunnel Mode

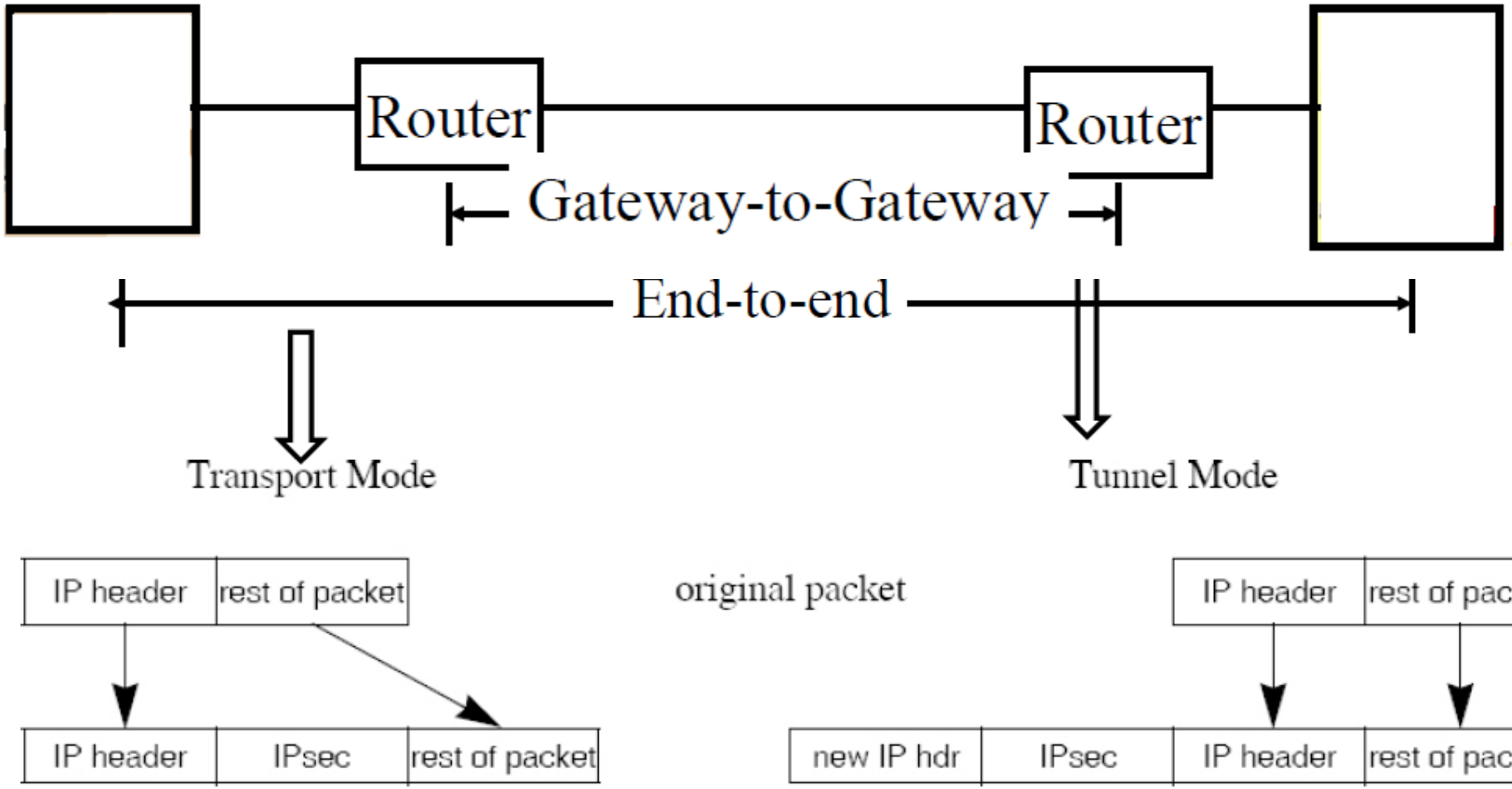
ESP in tunnel mode over the VPN also conceals the fact that Alice is talking to Bob

IPSec

- Secure IP: A series of proposals from IETF
- Separate Authentication and privacy
- Authentication Header (AH) ensures data integrity and data origin authentication
- Encapsulating Security Protocol (ESP) ensures confidentiality, data origin authentication, connectionless integrity, and anti-replay service

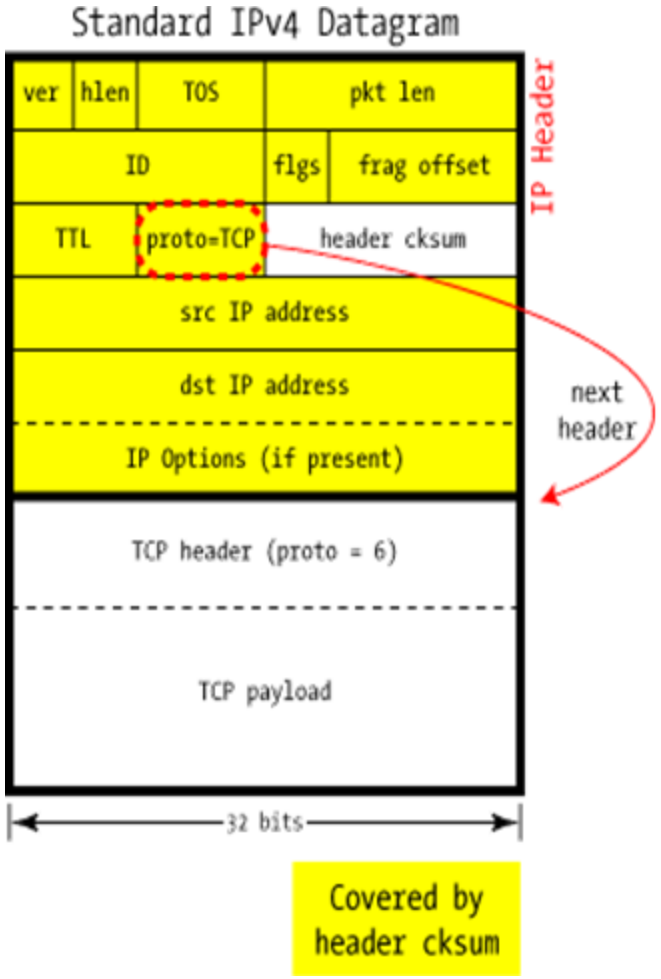


*Optional


Router
Router
Gateway-to-Gateway
End-to-end
Transport Mode
Tunnel Mode
IP header
rest of packet
IP header
IPsec
rest of packet
original packet
IP header
rest of packet
new IP hdr
IPsec
IP header
rest of packet

IPv4 Datagram

Some IP protocol codes	
Protocol code	Protocol Description
1	ICMP — Internet Control Message Protocol
2	IGMP — Internet Group Management Protocol
4	IP within IP (a kind of encapsulation)
6	TCP — Transmission Control Protocol
17	UDP — User Datagram Protocol
41	IPv6 — next-generation TCP/IP
47	GRE — Generic Router Encapsulation (used by PPTP)
50	IPsec: ESP — Encapsulating Security Payload
51	IPsec: AH — Authentication Header



Some fields

hlen: IP Header length, as a four-bit number of 32-bit words ranging from 0..15. A standard IPv4 header is always 20 bytes long (5 words), and IP Options — if any — are indicated by a larger **hlen** field up to at most 60 bytes. This header length never includes the size of payload or other headers that follow.

TOS: Type of Service- optimize for bandwidth, latency,low cost, reliability

Pkt_len : Overall packet length in bytes

ID: The ID field is related to packet fragmetation

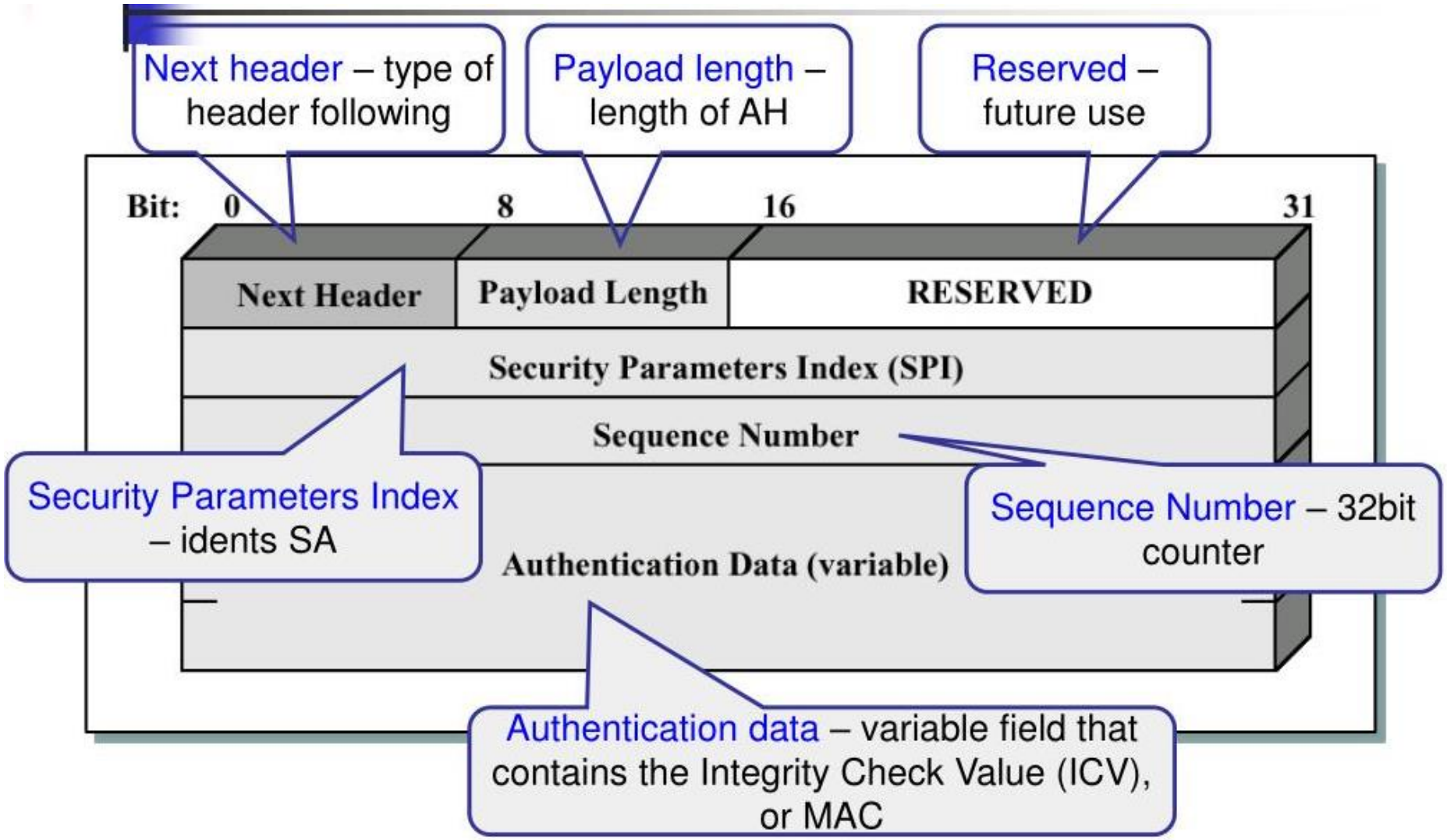
AH: Authentication Only

AH is used to authenticate — but not encrypt — IP traffic

Authentication is performed by computing a cryptographic hash-based message authentication code over nearly all the fields of the IP packet (excluding those which might be modified in transit, such as TTL or the header checksum), and stores this in a newly-added AH header and sent to the other end.

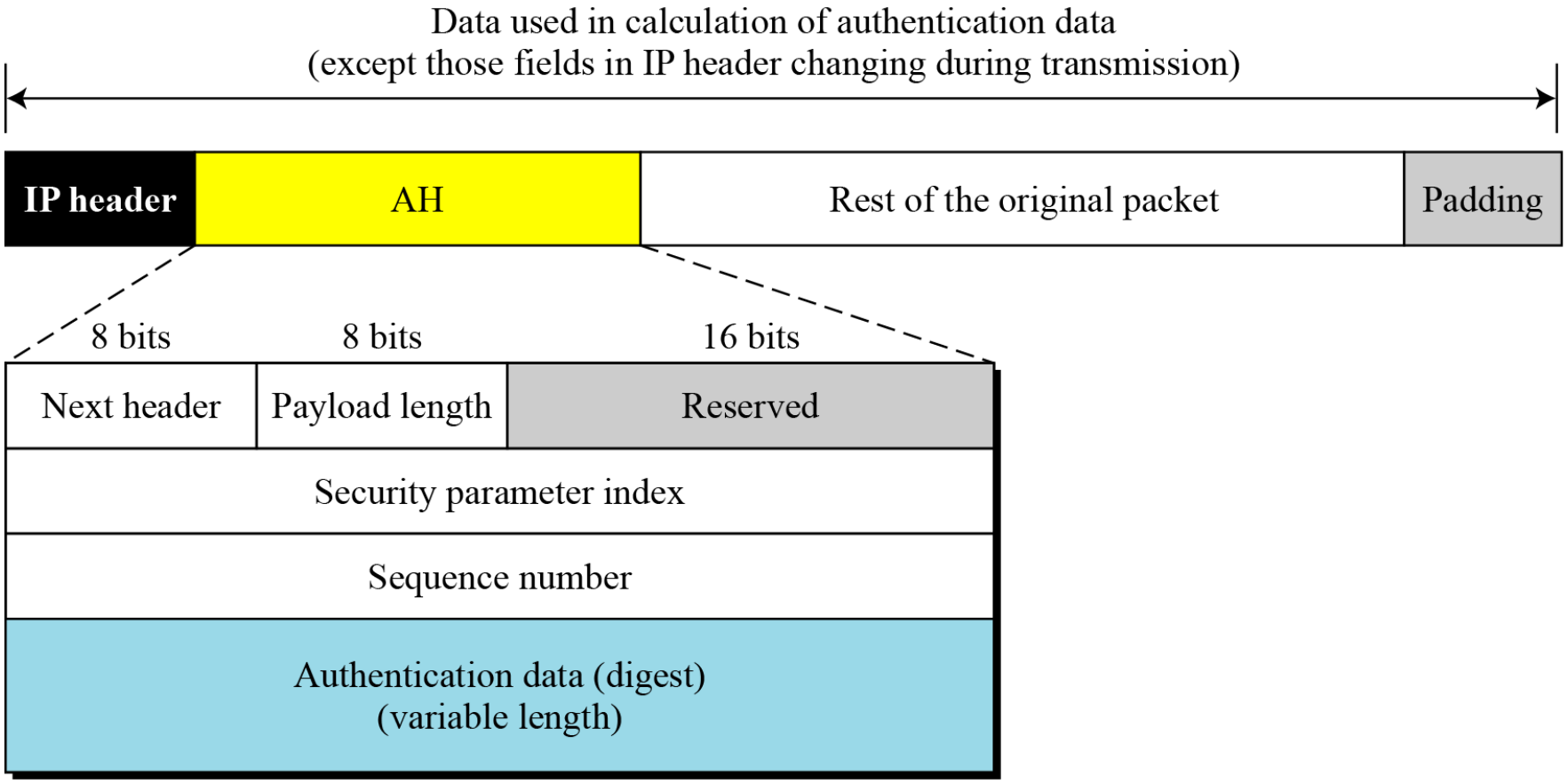
AH len: This defines the length, in 32-bit words, of the whole AH header, minus two words

Authentication Data: This is the Integrity Check Value calculated over the entire packet — including most of the headers — The recipient recomputes the same hash;


Next header – type of header following
Payload length – length of AH
Reserved – future use
Bit: 0
8
16
31
Next Header
Payload Length
RESERVED
Security Parameters Index (SPI)
Sequence Number
Authentication Data (variable)
Security Parameters Index – idents SA
Sequence Number – 32bit counter
Authentication data – variable field that contains the Integrity Check Value (ICV), or MAC

AH

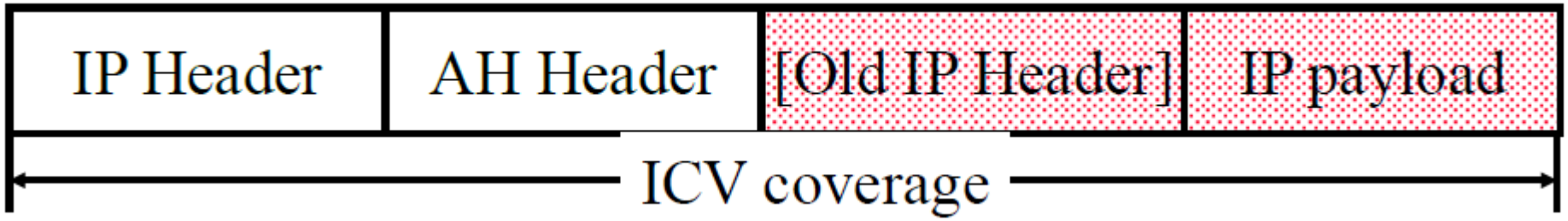
Figure Authentication Header (AH) protocol



AH

- Next Header = TCP=6, UDP=17, IP=4, AH=51 ⇒ Designed by IPv6 fans
- Payload Length = Length of ***AH*** in 32-bit words − 2 (for IPv4) =Length of AH in 64-bit words -1 (for IPv6)
- SPI = Identifies Security association (0=Local use, 1-255 reserved)
- Authentication data = Integrity Check Value

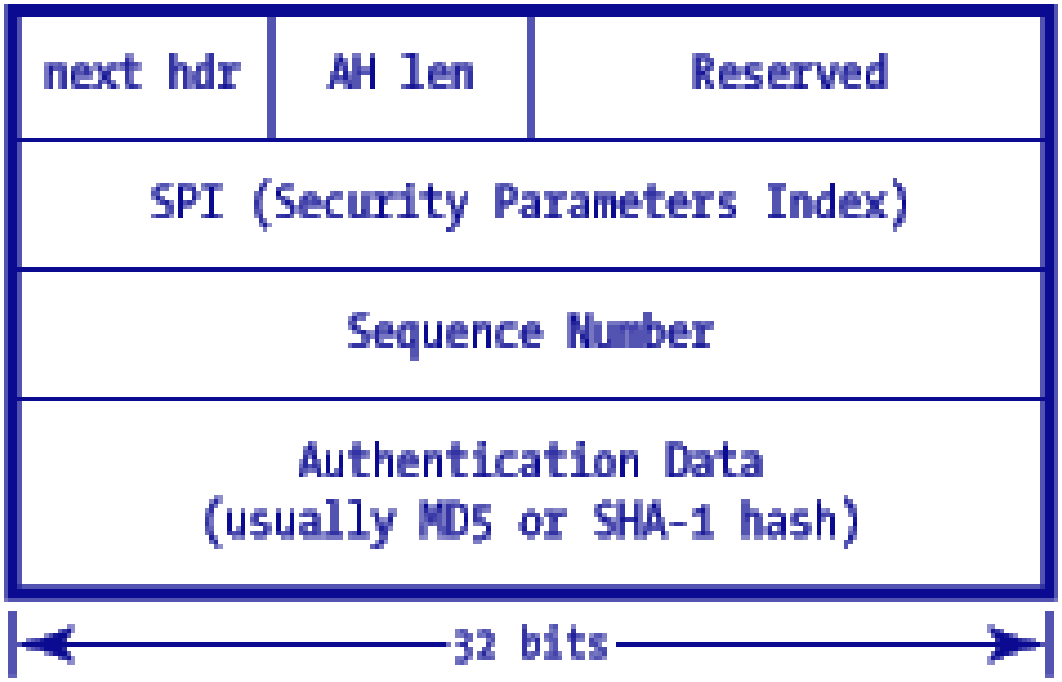
AH ICV Computation



The AH ICV is computed over:

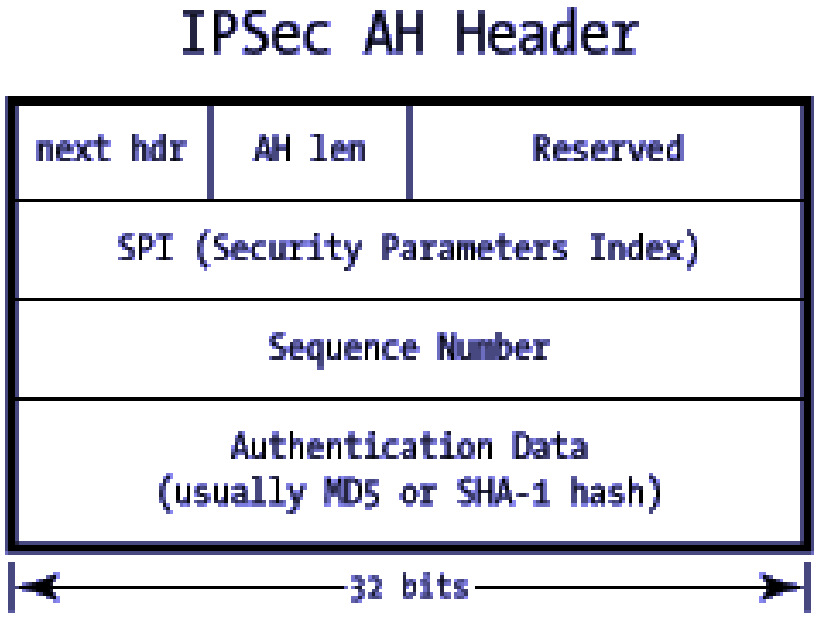
- IP header fields that are either immutable in transit or that are predictable in value upon arrival at the endpoint for the AH SA, e.g., source address (immutable), destination address with source routing (mutable but predictable)
- The AH header (Next Header, Payload Len, Reserved, SPI, Sequence Number, and the Authentication Data (which is set to zero for this computation), and explicit padding bytes (if any))
- The upper level protocol data, which is assumed to be immutable in transit

IPSec AH Header

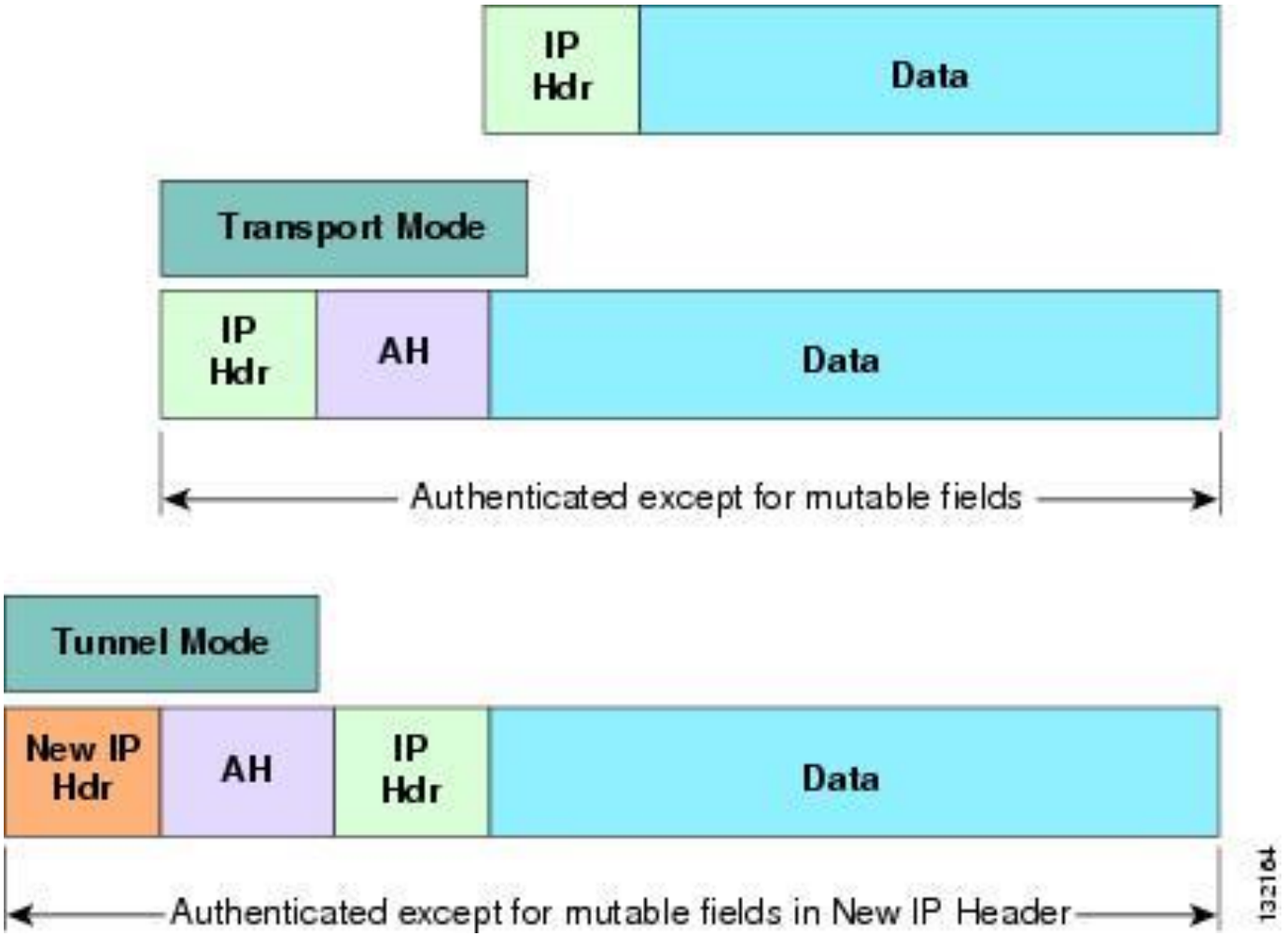


Transport Mode

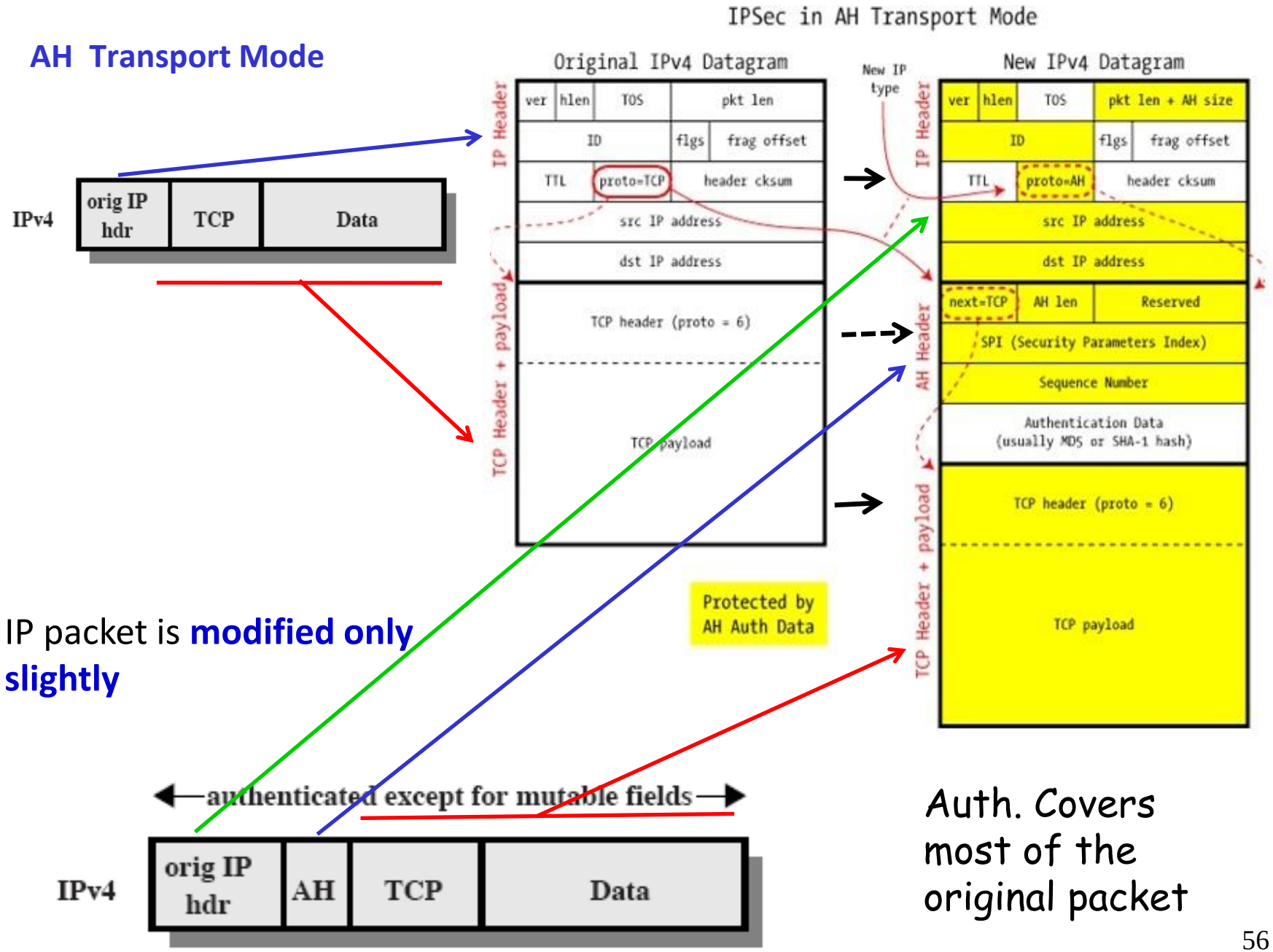
Transport Mode, which is used to protect an end-to-end conversation between two hosts.



In AH Transport Mode, the IP packet is **modified only slightly** to include the new AH header **between the IP** header **and the protocol payload** (TCP, UDP, etc.), and there is a shuffling of the protocol code that links the various headers together.


IP Hdr
Data
Transport Mode
IP Hdr
AH
Data
Authenticated except for mutable fields
Tunnel Mode
New IP Hdr
AH
IP Hdr
Data
Authenticated except for mutable fields in New IP Header
132164

AH Transport Mode



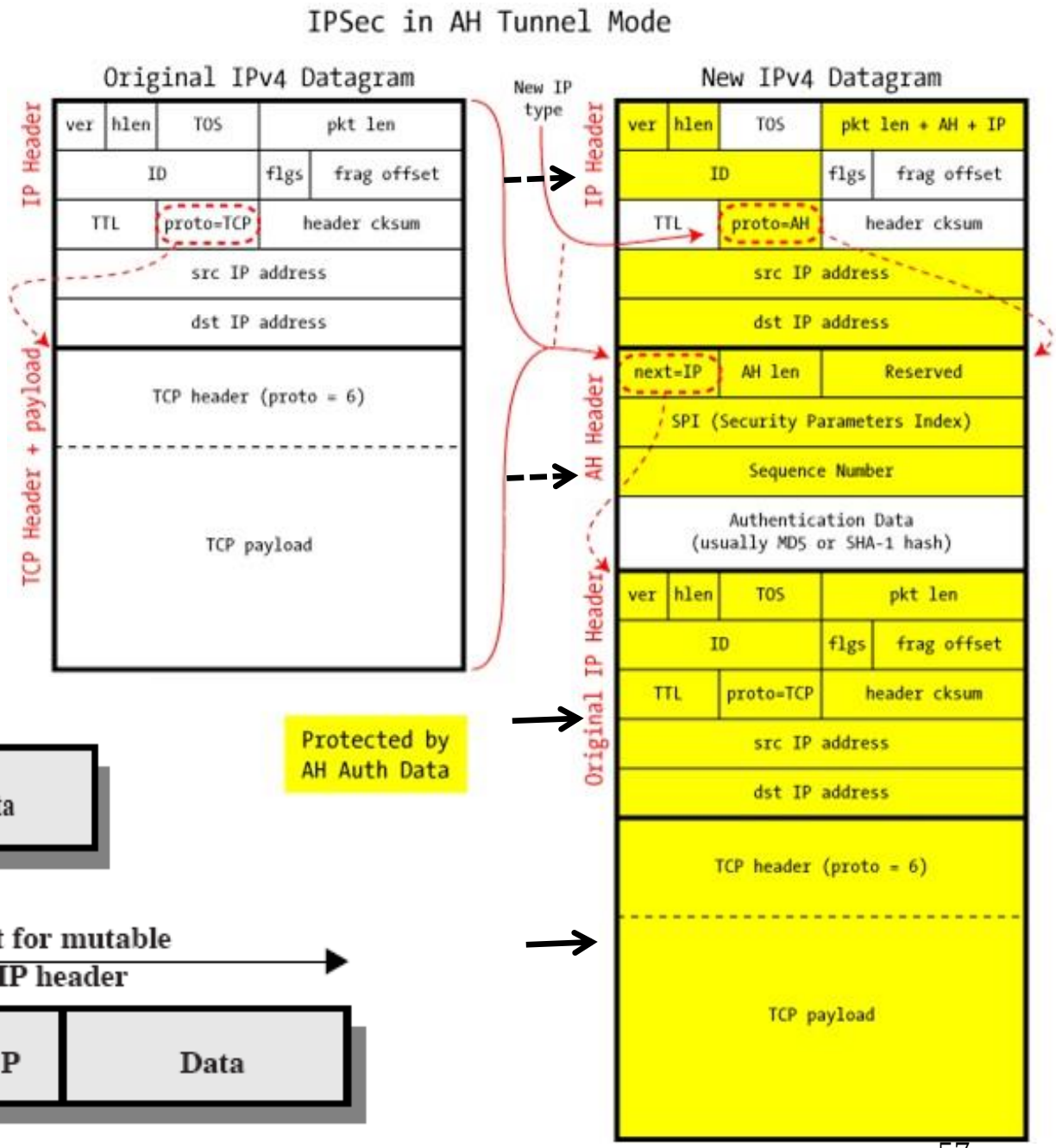
IP packet is **modified only slightly**

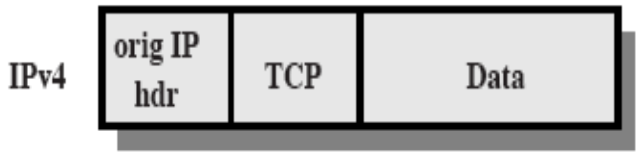
Auth. Covers most of the original packet

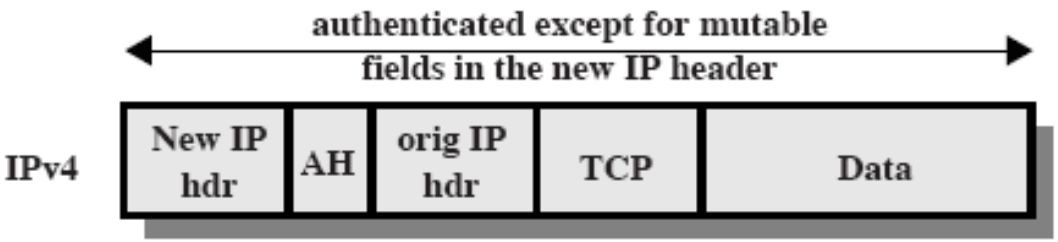
AH Tunnel Mode

entire IP packets are encapsulated inside another and delivered to the destination.

Auth. Cover entire original packet







Transport or Tunnel?

what distinguishes Transport mode from Tunnel mode **is the next header field** in the AH header.

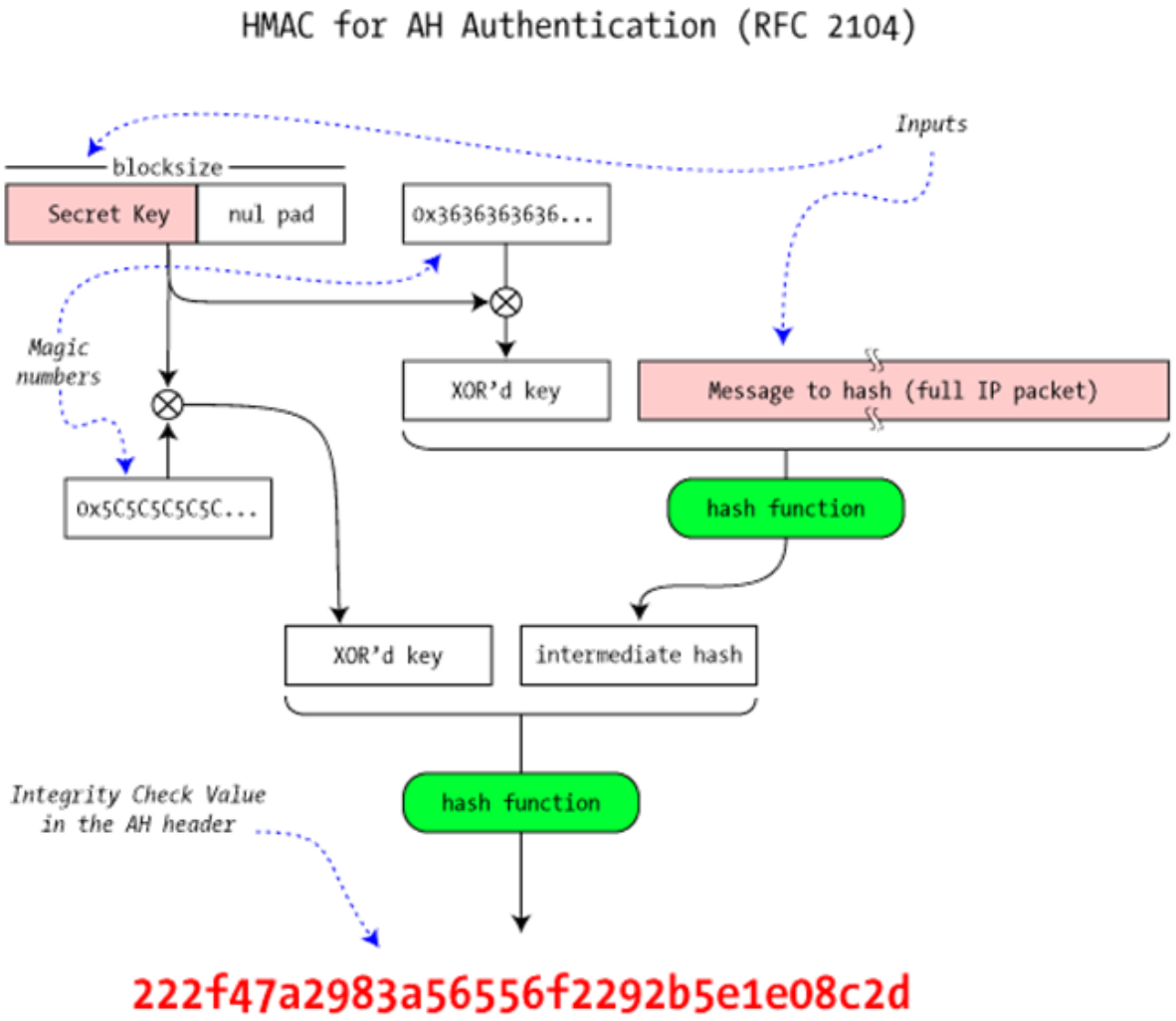
In tunnel mode next-header value is **IP**, it means that this packet encapsulates an entire IP datagram (including the independent source and destination IP addresses that allow separate routing after de-encapsulation).

Any other value (TCP, UDP, ICMP, etc.) means that it's Transport mode and is securing an endpoint-to-endpoint connection.

The top-level of the IP datagram is structured the same way regardless of mode, and intermediate routers treat all flavors IPsec/AH traffic identically without deeper inspection.

Authentication Algorithms

AH carries an Integrity Check Value in the Authentication Data portion of the header, and it's typically (but not always) built on top of standard cryptographic hash algorithms such as MD5 or SHA-1.



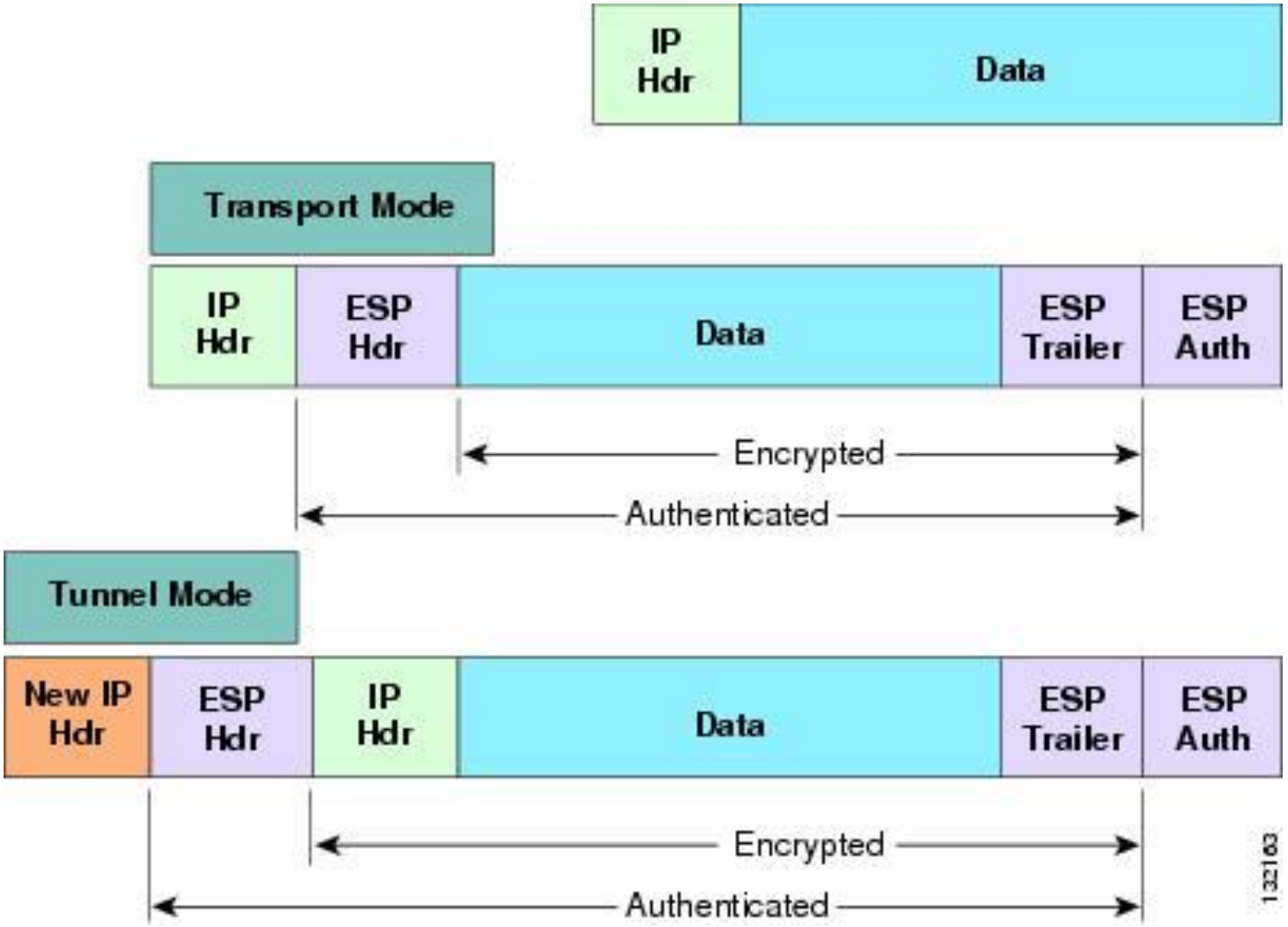
ESP — Encapsulating Security Payload

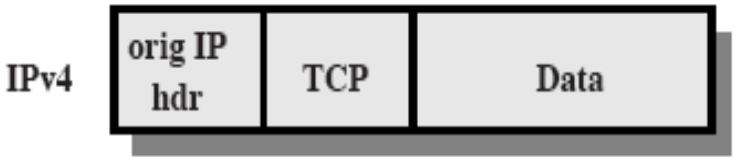
Adding encryption makes ESP a bit more complicated because encapsulation surrounds the payload rather than precedes it as with AH:

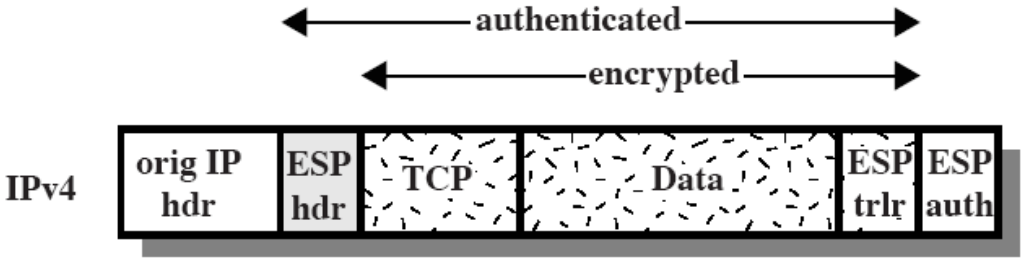
ESP includes header and trailer fields to support the encryption and optional authentication. It also provides Tunnel and Transport modes which are used in by-now familiar ways.

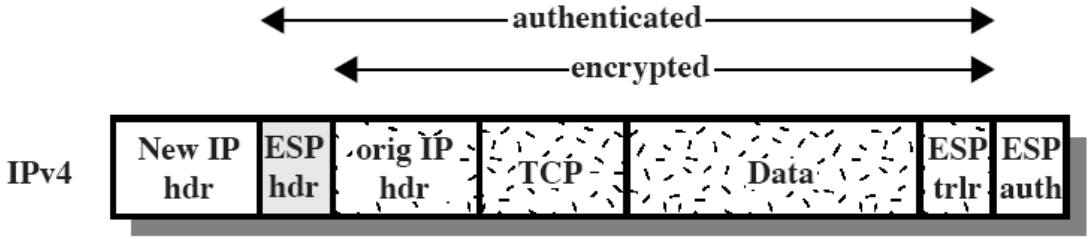
The IPsec RFCs don't insist upon any particular encryption algorithms, but we find DES, triple-DES, AES, and Blowfish in common use to shield the payload from prying eyes.

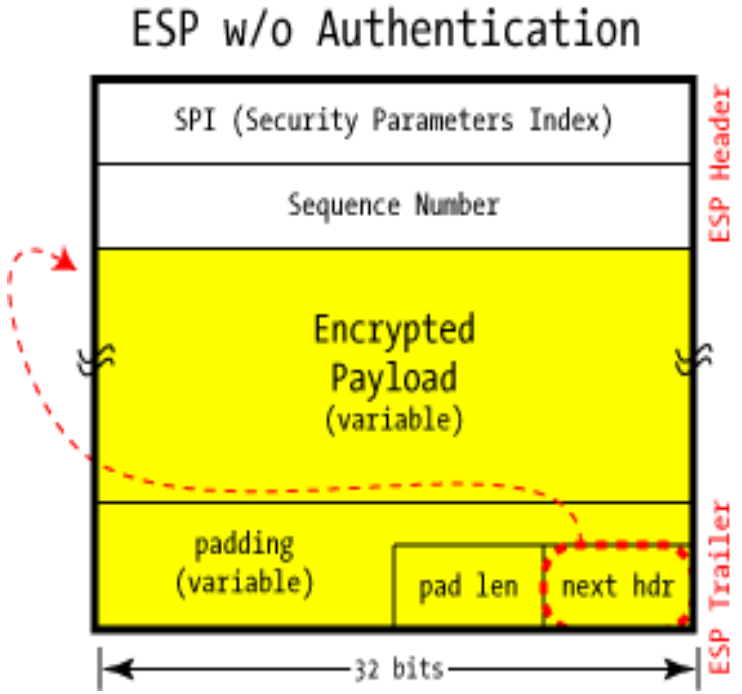
The algorithm used for a particular connection is specified by the Security Association and this SA includes not only the algorithm, but the key used.


IP Hdr
Data
Transport Mode
IP Hdr
ESP Hdr
Data
ESP Trailer
ESP Auth
Encrypted
Authenticated
Tunnel Mode
New IP Hdr
ESP Hdr
IP Hdr
Data
ESP Trailer
ESP Auth
Encrypted
Authenticated
132163


IPv4
orig IP hdr
TCP
Data


authenticated
encrypted
IPv4
orig IP hdr
ESP hdr
TCP
Data
ESP trlr
ESP auth


authenticated
encrypted
IPv4
New IP hdr
ESP hdr
orig IP hdr
TCP
Data
ESP trlr
ESP auth

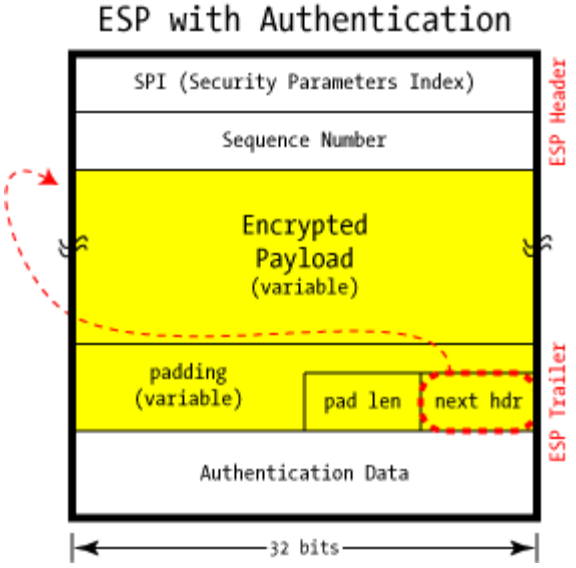

ESP w/o Authentication
SPI (Security Parameters Index)
Sequence Number
Encrypted Payload (variable)
padding (variable)
pad len
next hdr
32 bits
ESP Header
ESP Trailer

ESP — Encapsulating Security Payload (cont.)

It's possible to use ESP without any actual encryption (to use a NULL algorithm), which nonetheless structures the packet the same way. This provides no confidentiality, and it only makes sense if combined with ESP authentication.

Unlike AH, which provides a small header *before* the payload, ESP *surrounds* the payload it's protecting.

Padding is provided to allow block-oriented encryption algorithms room for multiples of their blocksize, and the length of that padding is provided in the pad len field.



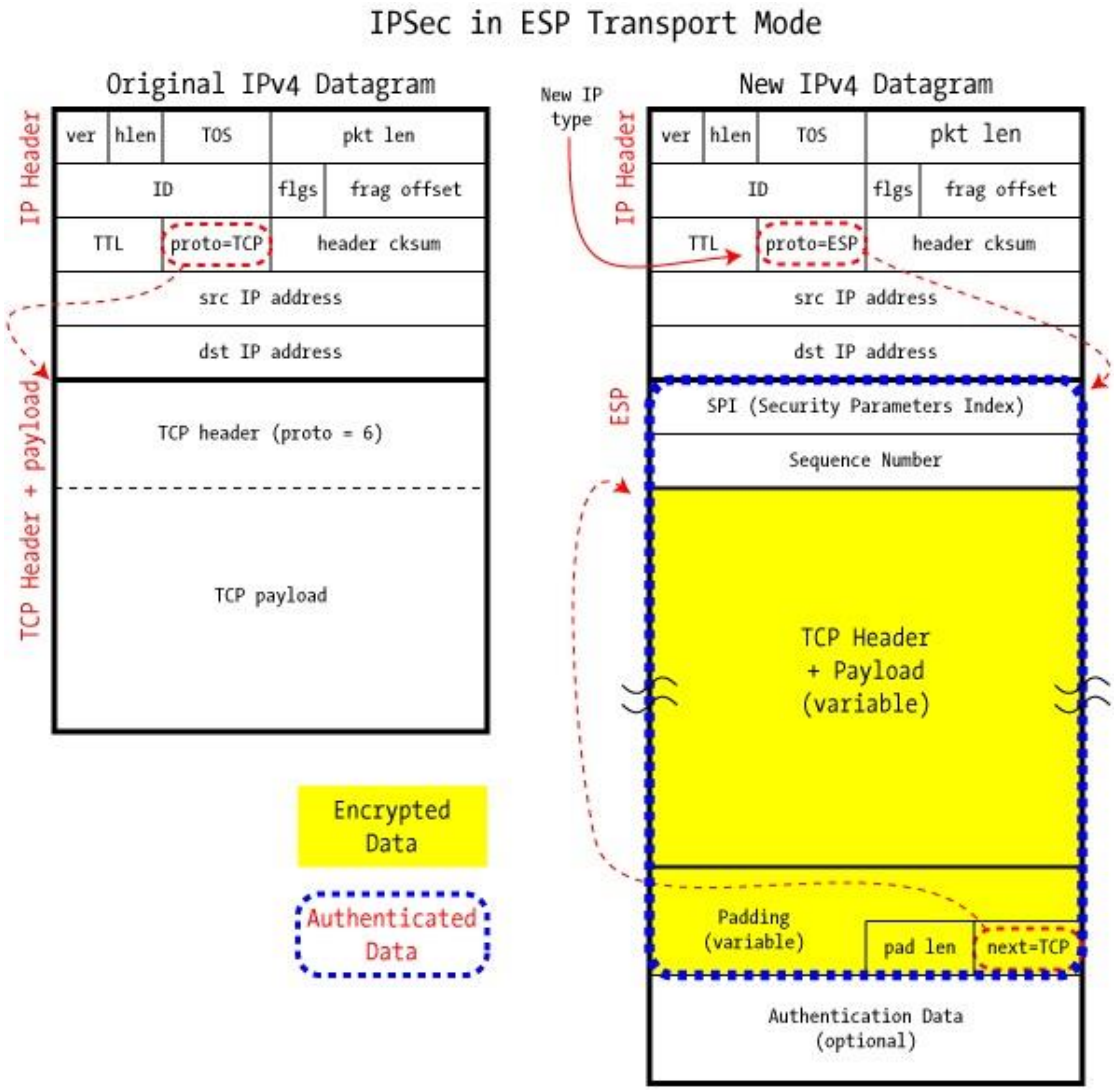
The **next hdr** field gives the type (IP, TCP, UDP, etc.) of the payload in the usual way, though it can be thought of as pointing "backwards" into the packet rather than forward as we've seen in AH.

In addition to encryption, ESP can also optionally provide authentication, with the same HMAC as found in AH. Unlike AH, however, this authentication is *only for the* ESP header and encrypted payload:

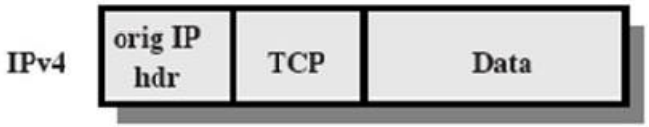
ESP in Transport Mode

As with AH, Transport Mode encapsulates just the datagram's payload and is designed strictly for host-to-host communications.

The original IP header is left in place (except for the shuffled Protocol field), and it means that — among other things — the source and destination IP addresses are unchanged.

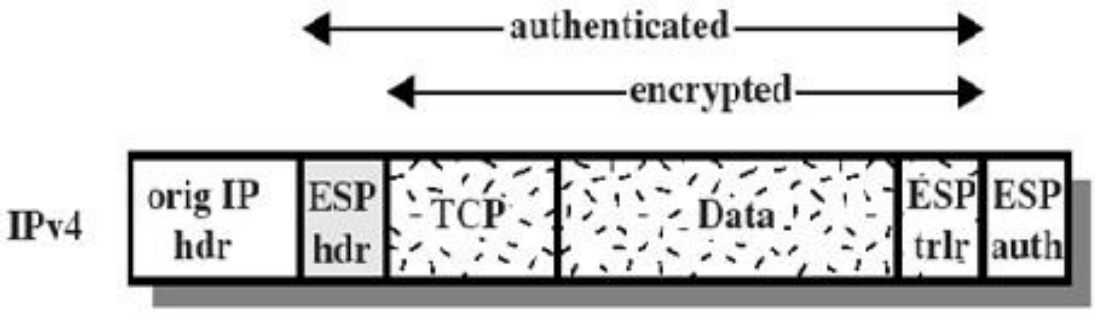


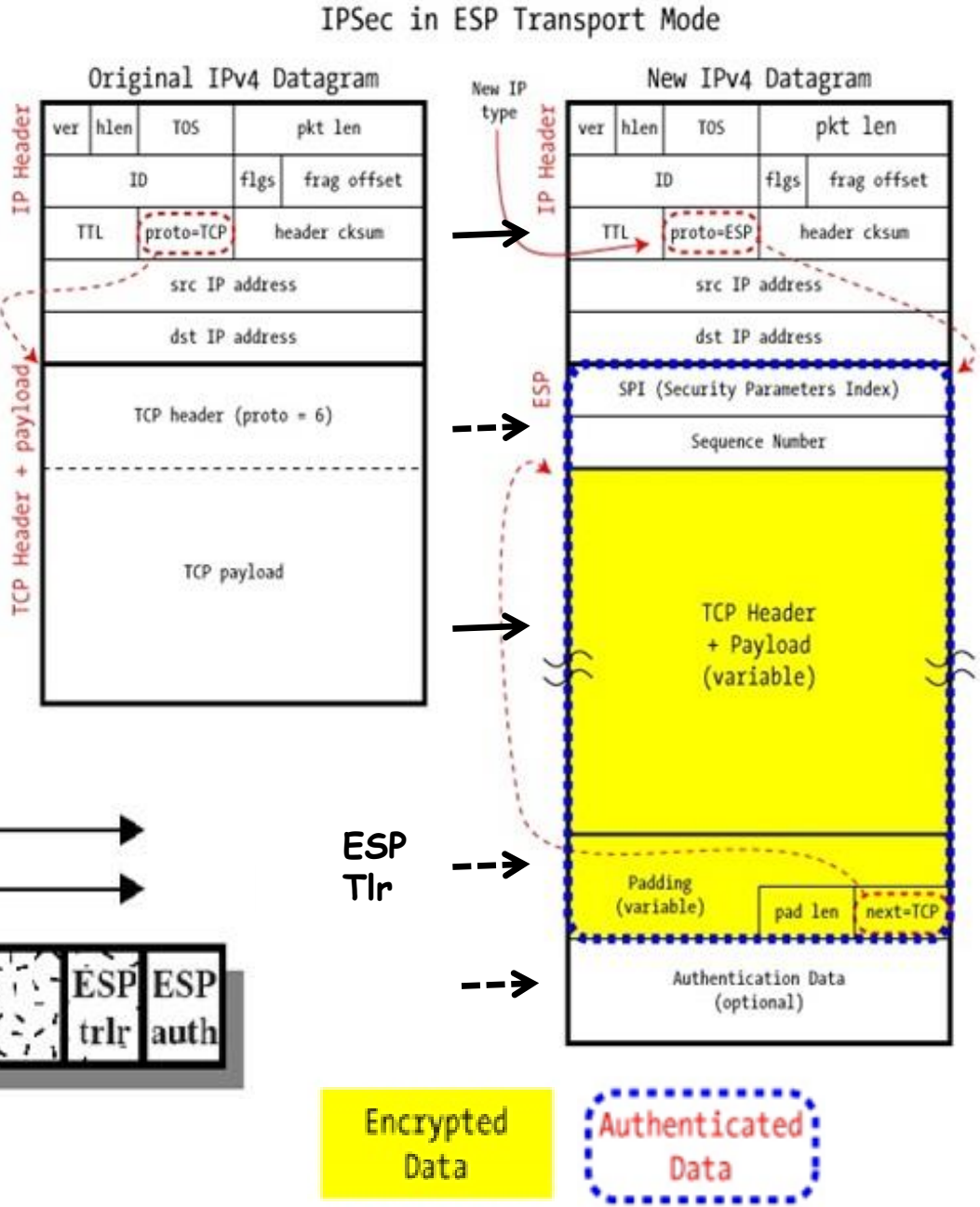
ESP in Transport Mode



Good for host to host traffic

IP packet is **modified only slightly** (proto fileld)



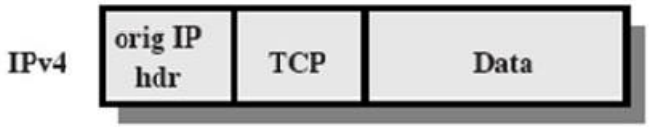


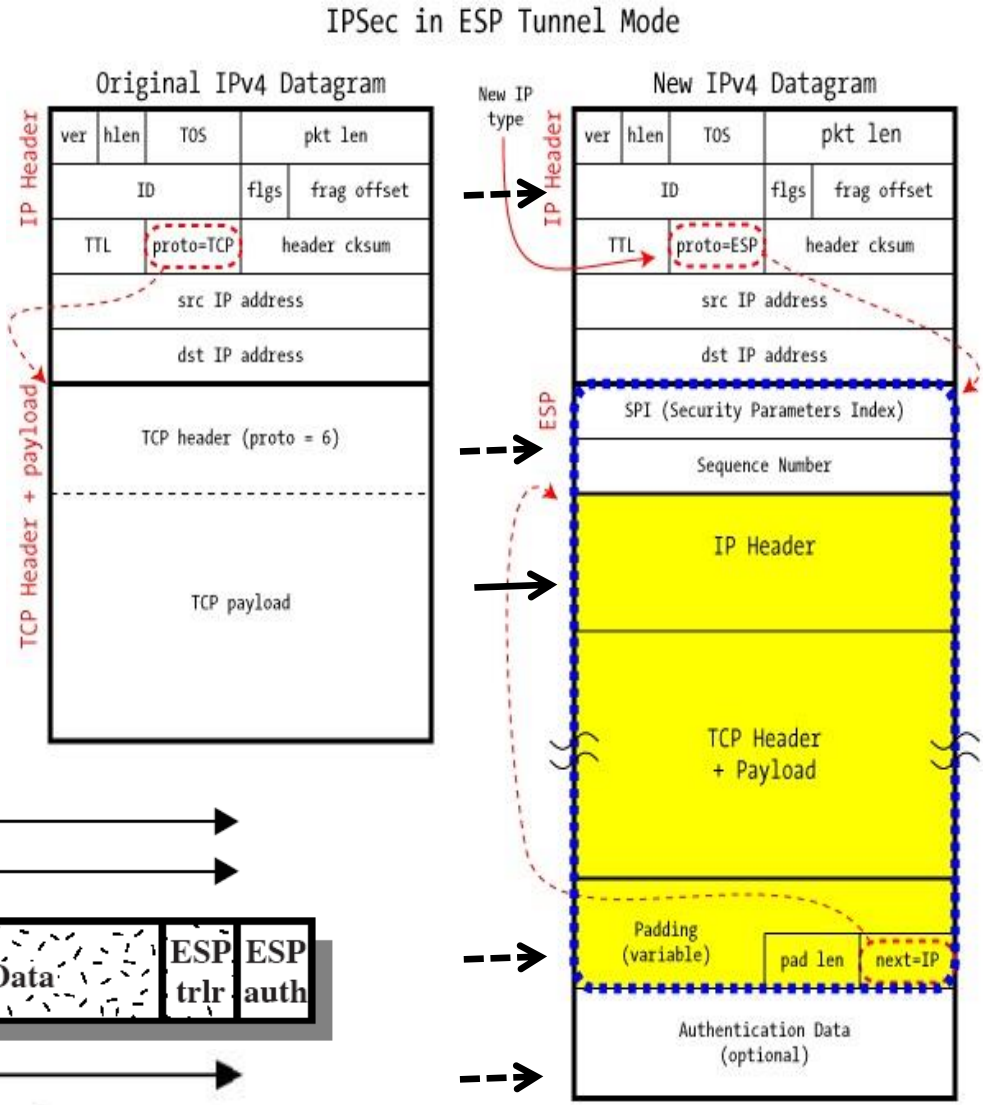
ESP in Tunnel Mode

ESP is in Tunnel mode encapsulates an entire IP datagram inside the encrypted shell:

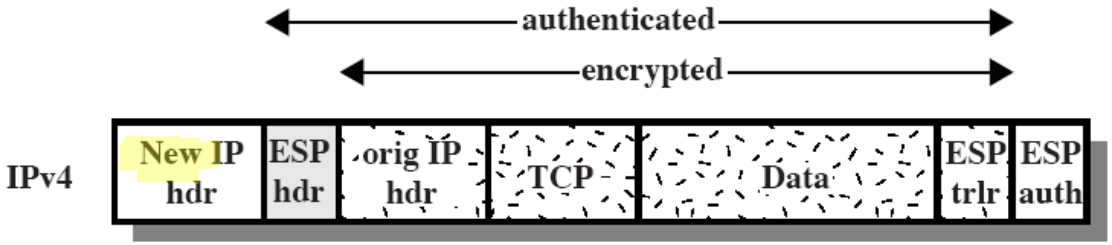
the fact that this is Tunnel mode (via next=IP) is part of the encrypted payload, and is simply not visible

ESP in Tunnel Mode is Good for VPNs, gateway to gateway security

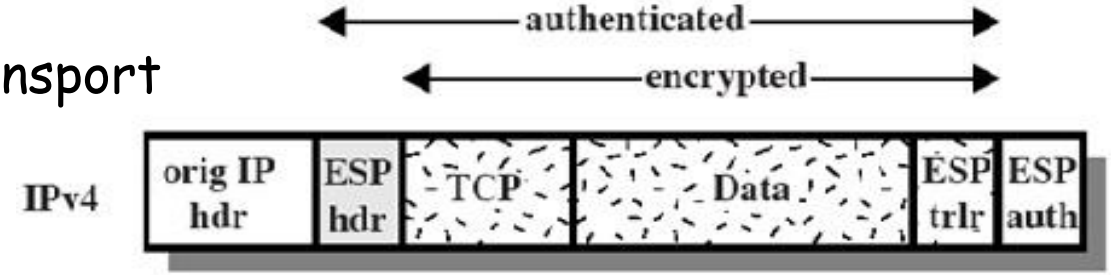




Tunnel



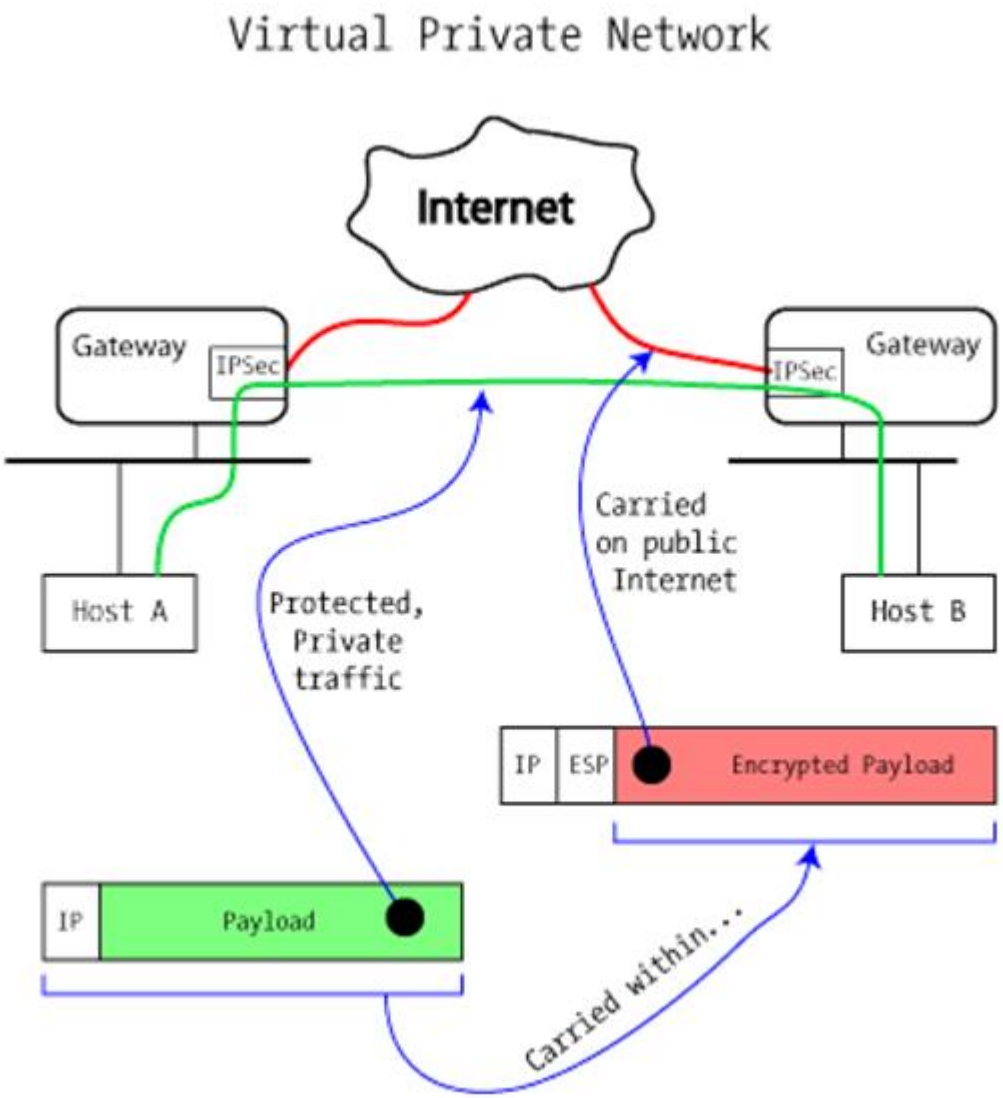
Transport



Putting it all together: Building a real VPN

The whole purpose of a Virtual Private Network is to join two trusted networks across an untrusted intermediate network, as is by stringing a very long Ethernet cable between the two.

This is commonly used to connect branch offices with company headquarters, allowing all users to share sensitive resources without fear of interception.



Building a real VPN (cont.)

Clearly, a secure VPN requires both authentication and encryption. We know that ESP is the only way to provide encryption, but ESP and AH both can provide authentication: which one do we use?

Instead (AH+ESP, as NAT problem) ESP+Auth is used in Tunnel mode to fully encapsulate the traffic on its way across an untrusted network, protected by both encryption and authentication in the same thing.

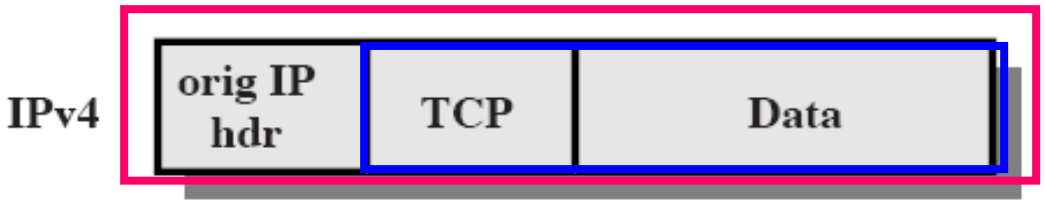
An outsider knows nothing about the actual traffic, even the type of encapsulated protocol — TCP, UDP, or ICMP which is hidden from outsiders.

The end-user hosts generally know nothing about the VPN or other security measures in place (Implemented in Gateways).

This packet-in-a-packet can actually be nested yet more levels: AH in ESP+AH.

Scope of ESP encryption and authentication

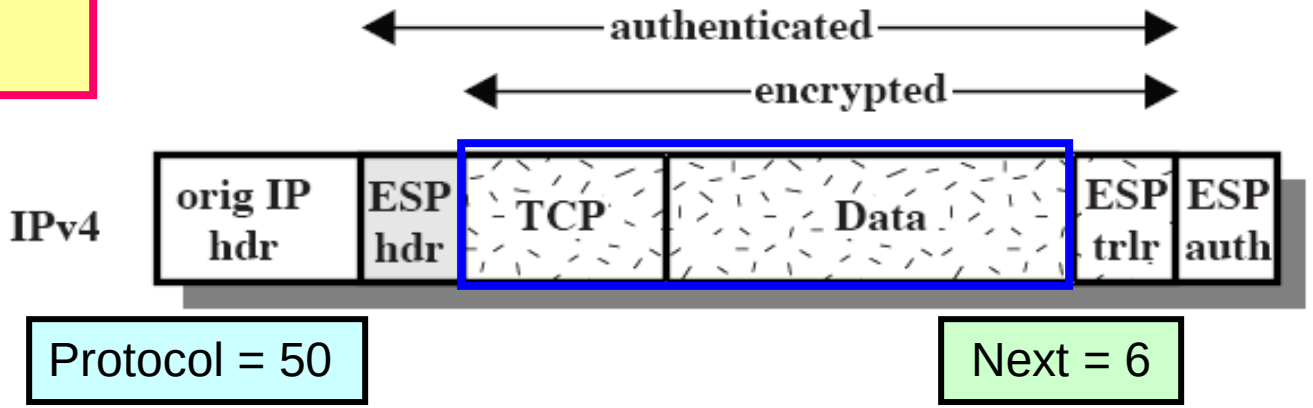
Original datagram



Protocol = 6

ESP authentication does not extend to the IP header

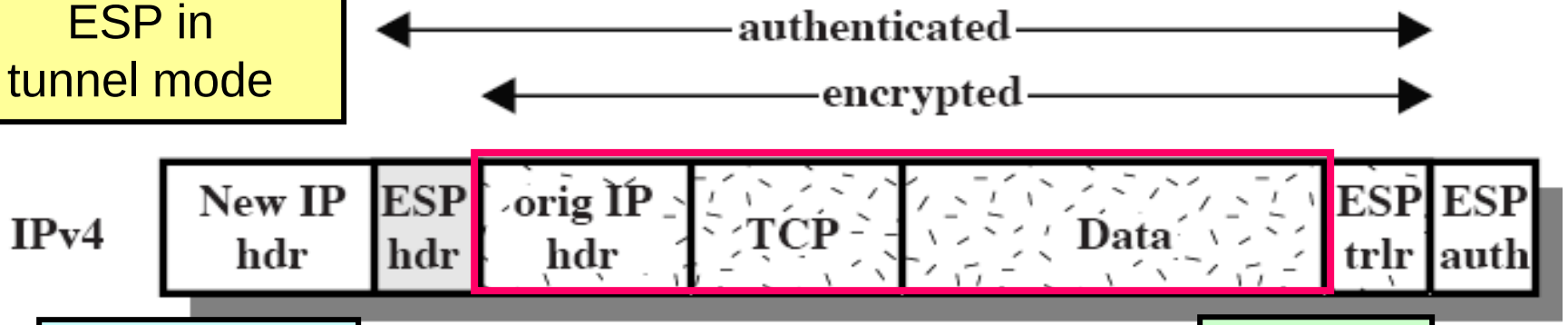
ESP in transport mode



Protocol = 50

Next = 6

ESP in tunnel mode



Protocol = 50

Next = 4

How to establish a Security Associations and the SPI

Clearly It seems self-evident that if two endpoints or gateways are going to establish a secure connection, some kind of shared secret is required to seed the authentication function and/or key the encryption algorithm.

This is specified by the Security Association (SA), a collection of connection-specific parameters, and each partner can have one or more Security Associations.

When a datagram arrives, three pieces of data are used to locate the correct SA inside the Security Associations Database (SADB):

Partner IP address
IPsec Protocol (ESP or AH)
Security Parameters Index

SADB

In many ways this triple can be likened to an IP socket, which is uniquely denoted by the remote IP address, protocol, and port number.

- Security Associations are one way, so a two-way connection (the typical case) requires at least two. Furthermore, each protocol (ESP/AH) has its own SA in each direction, so a full AH+ESP VPN requires four Security Associations. These are all kept in the Security Associations Database, entries include .

- AH: authentication algorithm
- AH: authentication secret
- ESP: encryption algorithm
- ESP: encryption secret key
- ESP: authentication enabled yes/no
- Many key-exchange parameters
- Routing restrictions
- IP filtering policy

SPD

Some implementations maintain the SPD (Security Policy Database) with command-line tools, others with a GUI, while others provide a web-based interface over the network.

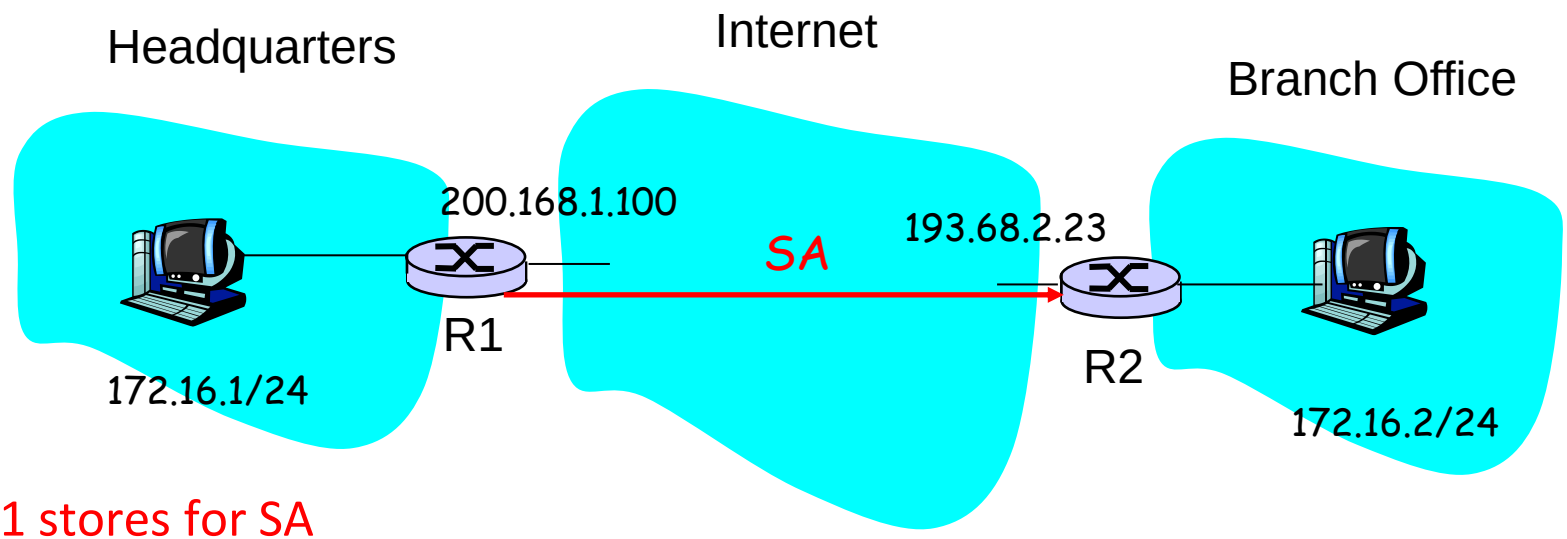
Key Management

IKE — Internet Key Exchange — exists to allow two endpoints to properly set up their Security Associations, including the secrets to be used.

IKE uses the ISAKMP (Internet Security Association Key Management Protocol) as a framework to support establishment of a security association compatible with both ends.

Multiple key-exchange protocols themselves are supported, with Oakley being the most widely used.

Example SA from R1 to R2



R1 stores for SA

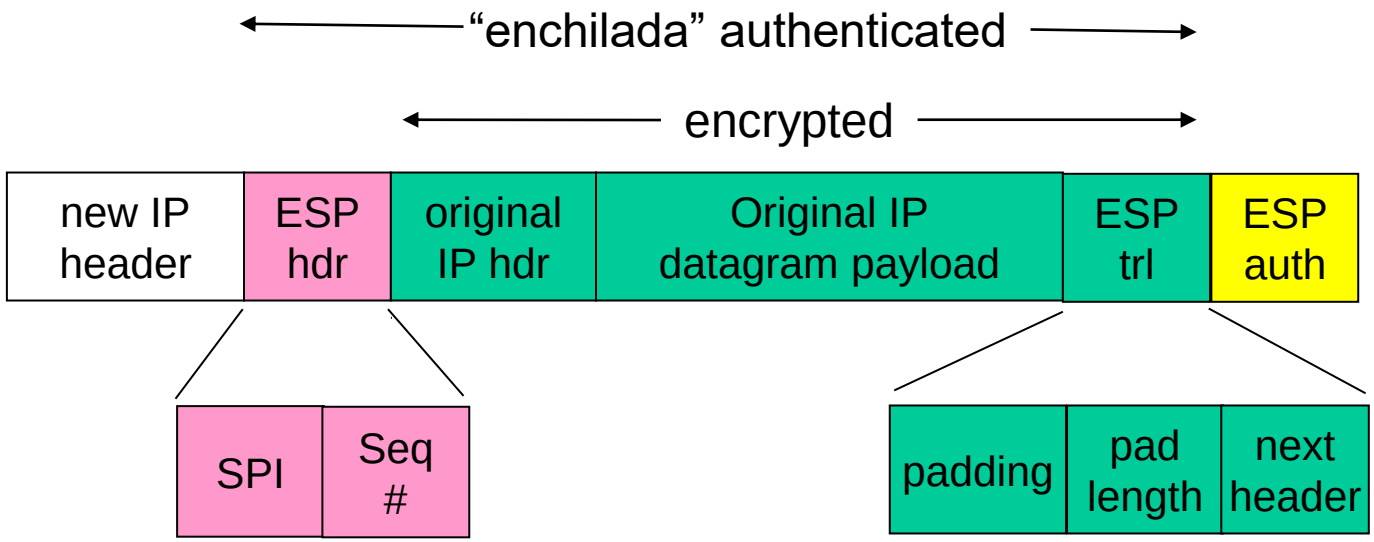
- 32-bit identifier for SA: Security Parameter Index (SPI)
- the origin interface of the SA (200.168.1.100)
- destination interface of the SA (193.68.2.23)
- type of encryption to be used (for example, 3DES with CBC)
- encryption key
- type of integrity check (for example, HMAC with with MD5)
- authentication key

Security Association Database (SAD)

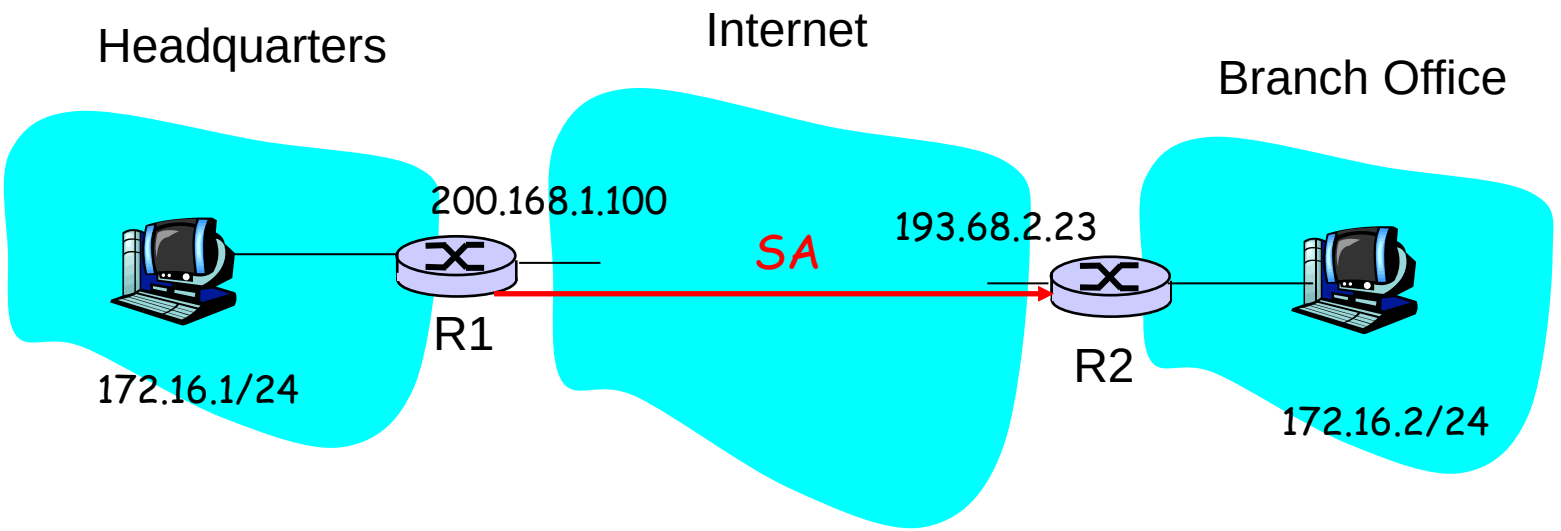
- Endpoint holds state of its SAs in a SAD, where it can locate them during processing.
- With n salespersons, 2 + 2n SAs in R1's SAD
- When sending IPsec datagram, R1 accesses SAD to determine how to process datagram.
- When IPsec datagram arrives to R2, R2 examines SPI in IPsec datagram, indexes SAD with SPI, and processes datagram accordingly.

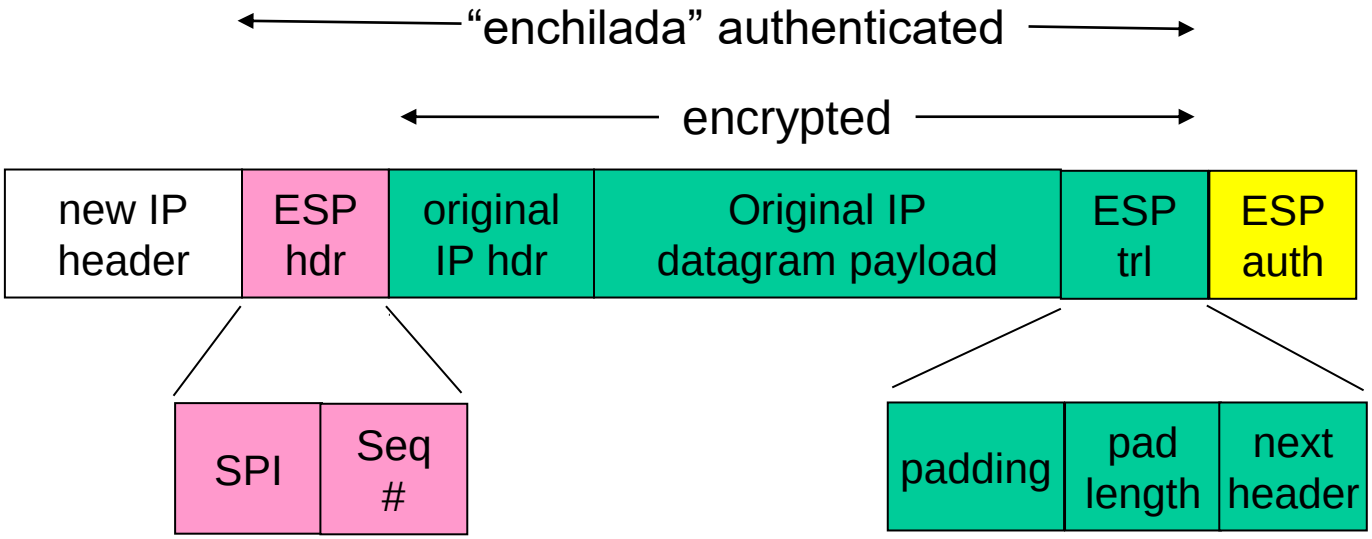
IPsec datagram

Focus for now on tunnel mode with ESP



What happens?





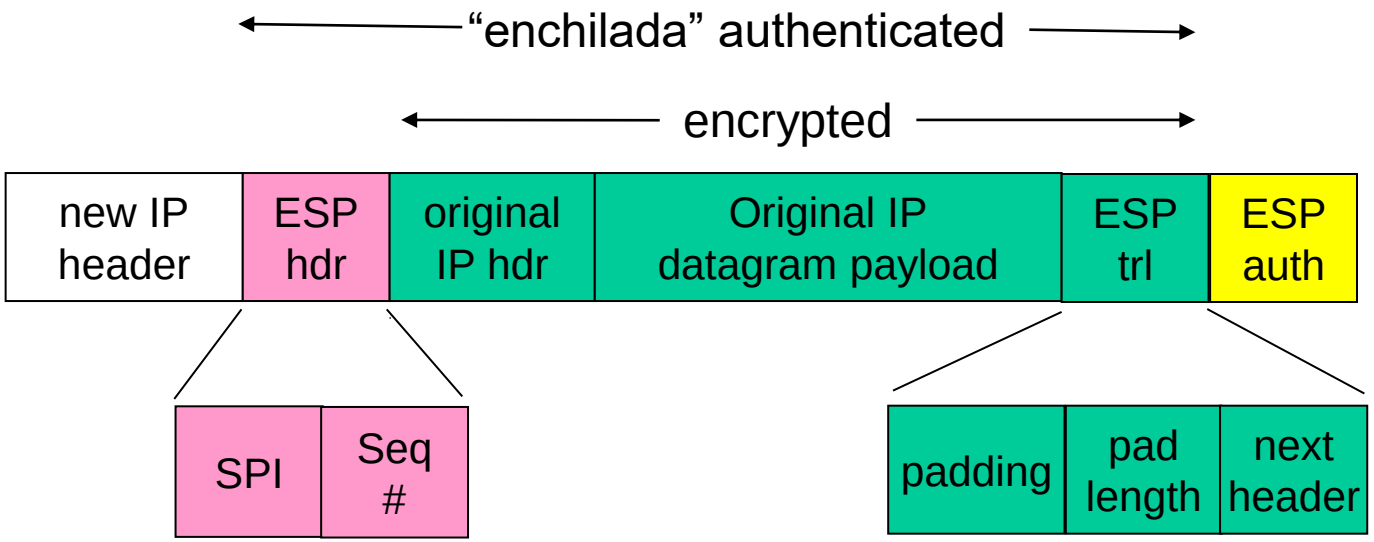
R1 converts original datagram into IPsec datagram

- Appends to back of original datagram (which includes original header fields!) an "ESP trailer" field.
- Encrypts result using algorithm & key specified by SA.
- Appends to front of this encrypted quantity the "ESP header, creating "enchilada".
- Creates authentication MAC over the *whole enchilada*, using algorithm and key specified in SA;
- Appends MAC to back of enchilada, forming *payload*;
- Creates brand new IP header, with all the classic IPv4 header fields, which it appends before payload.

R2 recovers original datagram from IPsec datagram

- When R2 receives the IPsec datagram, R2 observes that the destination IP address of the datagram is R2 itself. R2 therefore processes the datagram.
- Because the protocol field (in the left-most IP header) is 50, R2 sees that it should apply IPsec ESP processing to the datagram.
- First, peering into the enchilada, R2 uses the SPI to determine to which SA the datagram belongs.
- Second, it calculates the MAC of the enchilada and verifies that the MAC is consistent with the value in the ESP MAC field.
- Third, it checks the sequence-number field to verify that the datagram is fresh (and not a replayed datagram).
- Fourth, it decrypts the encrypted unit using thedecryption algorithm and key associated with the SA.
- Fifth, it removes padding and extracts the original, vanilla IP datagram.
- And finally, sixth, it forwards the original datagram into the branch office network towards its ultimate destination.

Inside the enchilada:

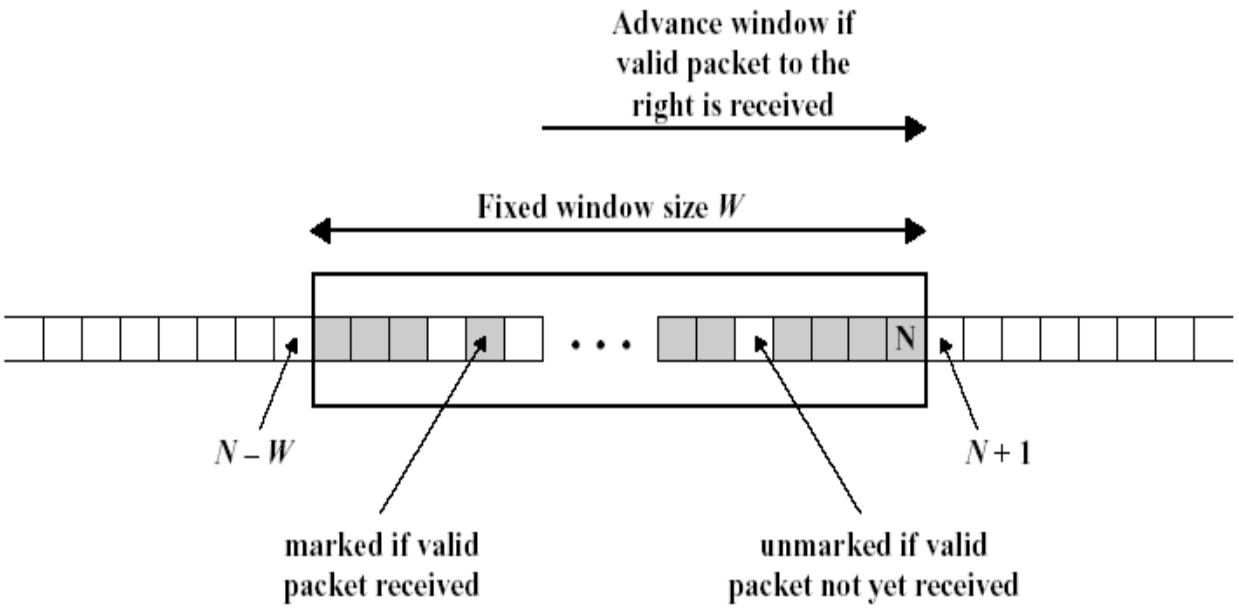


- ESP trailer: Padding for block ciphers
- ESP header:
  - SPI, so receiving entity knows what to do
  - Sequence number, to thwart replay attacks
- MAC in ESP auth field is created with shared secret key

IPsec sequence numbers

- For new SA, sender initializes seq. # to 0
- Each time datagram is sent on SA:
  - Sender increments seq # counter
  - Places value in seq # field
- Goal:
  - Prevent attacker from replaying a packet
    - Receipt of duplicate, authenticated IP packets may disrupt service
- Method:
  - Destination checks for duplicates
  - But doesn't keep track of ALL received packets; instead uses a window

Algorithm at receiver



N is highest sequence # rcvd.

Default W=64

1. If rcvd packet falls in window, packet is new, and MAC is valid ➜ slot in window marked
2. If rcvd packet is to right of window, MAC is valid ➜ window advanced & right-most slot marked
3. If rcvd packet is left of window, or already marked, or MAC not valid ➜ packet is discarded

How and What? (SPD)

1. How does R1 know whether it should be converted to an IPsec datagram?

2. And if it is to be processed by IPsec, how does R1 know which SA (of many SAs in its SAD) should be used to construct the IPsec datagram?

Ans

1. Along with a SAD, the Ipsec entity also maintains another data structure called the Security Policy Database (SPD).
2. The SPD indicates what types of datagrams (as a function of source IP address, destination IP address, and protocol type) are to be IPsec processed;
3. And for those that are to be IPsec processed, which SA should be used.

Security Policy Database (SPD)

- Policy: For a given datagram, sending entity needs to know if it should use IPsec.
- Needs also to know which SA to use
  - May use: source and destination IP address; protocol number.
- Info in SPD indicates "what" to do with arriving datagram;
- Info in the SAD indicates "how" to do it.

Focus on an **outbound** IP datagram crossing the boundary between an intranet and the Internet.

How is it decided what security processes are applied to this datagram?

This is a ***policy*** decision by administration.

The decision for each category of traffic is entered into a ***Security Policy Database.***

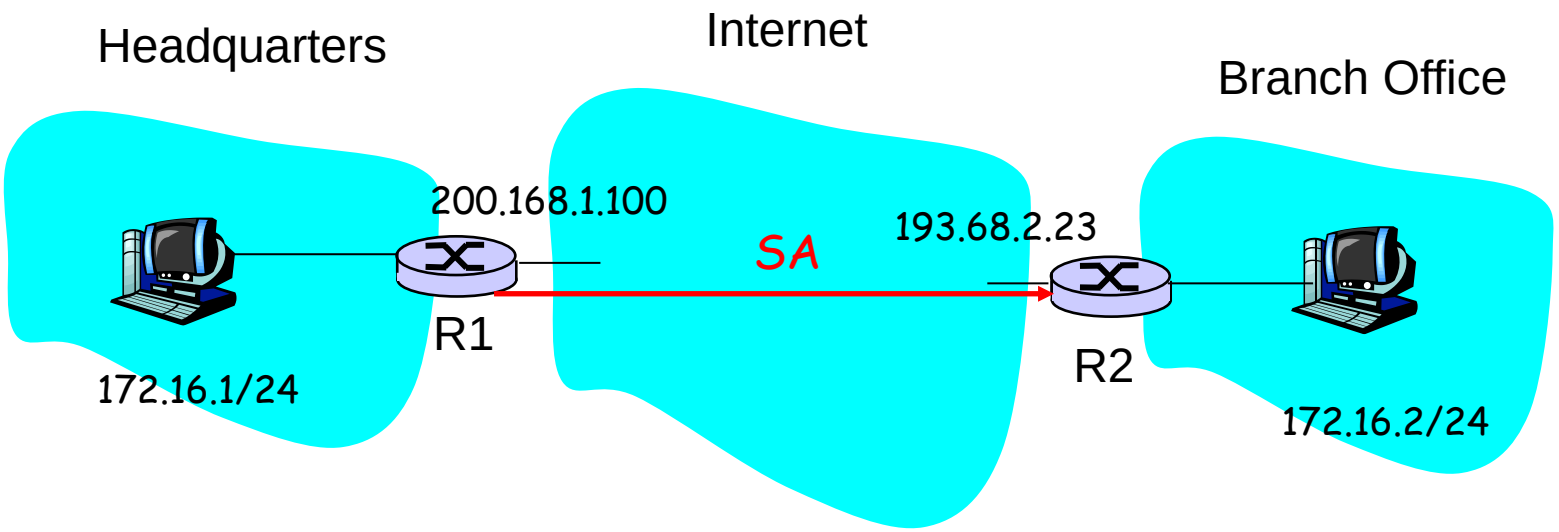
The menu of available processes is collected into a ***Security Association Database.***

Adopt 2-step process - entries in the Security Policy Database point to entries in the Security Association Database.

Summary of IPsec Services

1. Let us examine these services from the perspective of an attacker , say Turdy, who is a woman-in-the-middle, sitting somewhere on the path between R1 and R2
2. Trudy does not know the authentication and encryption keys used by the SA.
3. Trudy cannot see the original datagram.
4. Second, suppose Trudy can not to tamper with a datagram in the SA by flipping some of its bits.
5. Third, suppose Trudy tries to masquerade as R1, creating a IPsec datagram with source 200.168.1.100 and destination 193.68.2.23. Trudy's attack will be futile, as this datagram will again fail the integrity check at R2.
6. Finally, because IPsec includes sequence numbers, Trudy will not be able create a successful replay attack.

Linux example: ESP in tunnel mode



- In each host, create config file:
  - /etc/setkey.conf
- Execute setkey command in both hosts, which reads the setkey.conf file
  - setkey –f /etc/setkey.conf
  - Creates SAD and SPD databases

setkey.conf for R1

```
# Flush the SAD and SPD
flush;
spdflush;

# SAs encrypt w/ 192 bit keys & auth w/ 128 bit keys

Add 200.168.1.100 193.68.2.23 esp 0x201 -m tunnel -E 3des-cbc
0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831 -A
hmac-md5 0xc0291ff014dccdd03874d9e8e4cdf3e6;

Add 193.68.2.23 200.168.1.100  esp 0x301 -m tunnel -E 3des-cbc
0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df -A
hmac-md5 0x96358c90783bbfa3d7b196ceabe0536b;

# Security policies

spdadd 172.16.1.0/24 172.16.2.0/24 any -P out ipsec
esp/tunnel/ 200.168.1.100 - 193.68.2.23 /require;

spdadd 172.16.2.0/24 172.16.1.0/24 any -P in ipsec
esp/tunnel/ 193.68.2.23 - 200.168.1.100 /require;
```

ESP protocol

SPI

2 SAs added to SAD

apply to all packets

2 policies added to SPD

Another Example: AH in Transport Mode between R1 and R2

```
# Flush the SAD and SPD
flush;
spdflush;

# AH SAs using 128 bit long keys
Add 200.168.1.100  193.68.2.23 ah 0x200 -A hmac-md5
0xc0291ff014dccdd03874d9e8e4cdf3e6;
Add 193.68.2.23  200.168.1.100 ah 0x300 -A hmac-md5
0x96358c90783bbfa3d7b196ceabe0536b;

# Security policies
Spdadd 200.168.1.100  193.68.2.23 any -P out ipsec
     ah/transport//require;
Spdadd 193.68.2.23  200.168.1.100 any -P in ipsec
     ah/transport//require;
```

2 SAs added to SAD

2 policies added to SPD

Possible encryption algorithms

- DES
- 3DES
- AES
- RC5
- IDEA
- 3-IDEA
- CAST
- Blowfish
- ….

IPsec Security

- Suppose Trudy sits somewhere between R1 and R2. She doesn't know the keys.
  - Will Trudy be able to see contents of original datagram? How about source, dest IP address, transport protocol, application port?
  - Flip bits without detection?
  - Masquerade as R1 using R1's IP address?
  - Replay a datagram?

IKE: Key Management in IPsec

--Such "manual keying" is clearly impractical for a large VPN, which may consist of hundreds or even thousands of IPsec routers and hosts.

--Large, geographically distributed deployments require an automated mechanism for creating the SAs.

--IPsec does this with the Internet Key Exchange (IKE) protocol, specified in RFC 5996.

----Each IPsec entity has a certificate, which includes the entity's public key.

----As with SSL, the IKE protocol has the two entities exchange certificates, negotiate authentication and encryption algorithms, and securely exchange key material for creating session keys in the IPsec SAs.

Internet Key Exchange

- In previous examples, we manually established IPsec SAs in IPsec endpoints:

Example SA

SPI: 12345

Source IP: 200.168.1.100

Dest IP: 193.68.2.23

Protocol: ESP

Encryption algorithm: 3DES-cbc

HMAC algorithm: MD5

Encryption key: 0x7aeaca…

HMAC key:0xc0291f…

- Such manually keying is impractical for large VPN with, say, hundreds of sales people.
- Instead use *IPsec IKE (Internet Key Exchange)*

IKE: PSK and PKI

- Authentication (proof who you are) with either
  - pre-shared secret (PSK) or
  - with PKI (pubic/private keys and certificates).
- With PSK, both sides start with secret:
  - then run IKE to authenticate each other and to generate IPsec SAs (one in each direction), including encryption and authentication keys
- With PKI, both sides start with public/private key pair and certificate.
  - run IKE to authenticate each other and obtain IPsec SAs (one in each direction).
  - Similar with handshake in SSL.

Summary of relationship between SPD, IKE, and SAD:

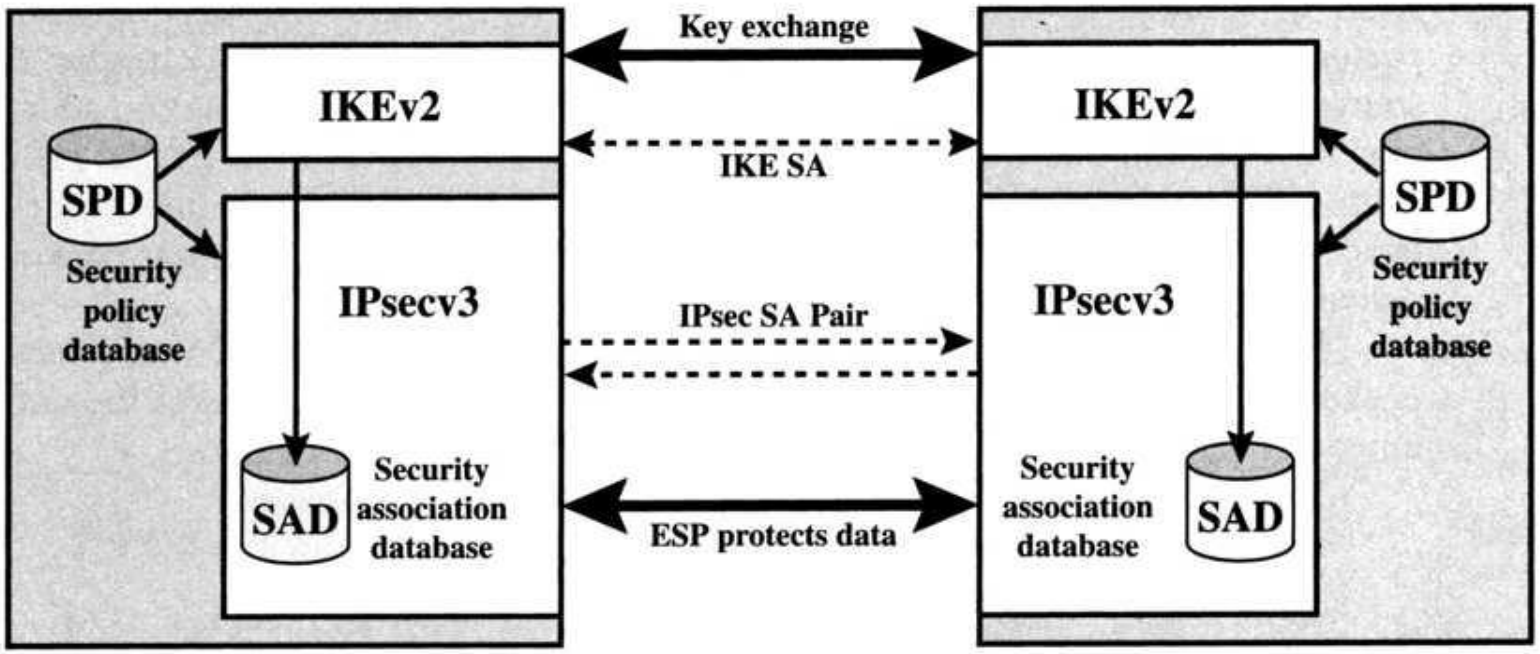


Figure 8.2 IPsec Architecture

Note that SA must first be established between the two IKEs

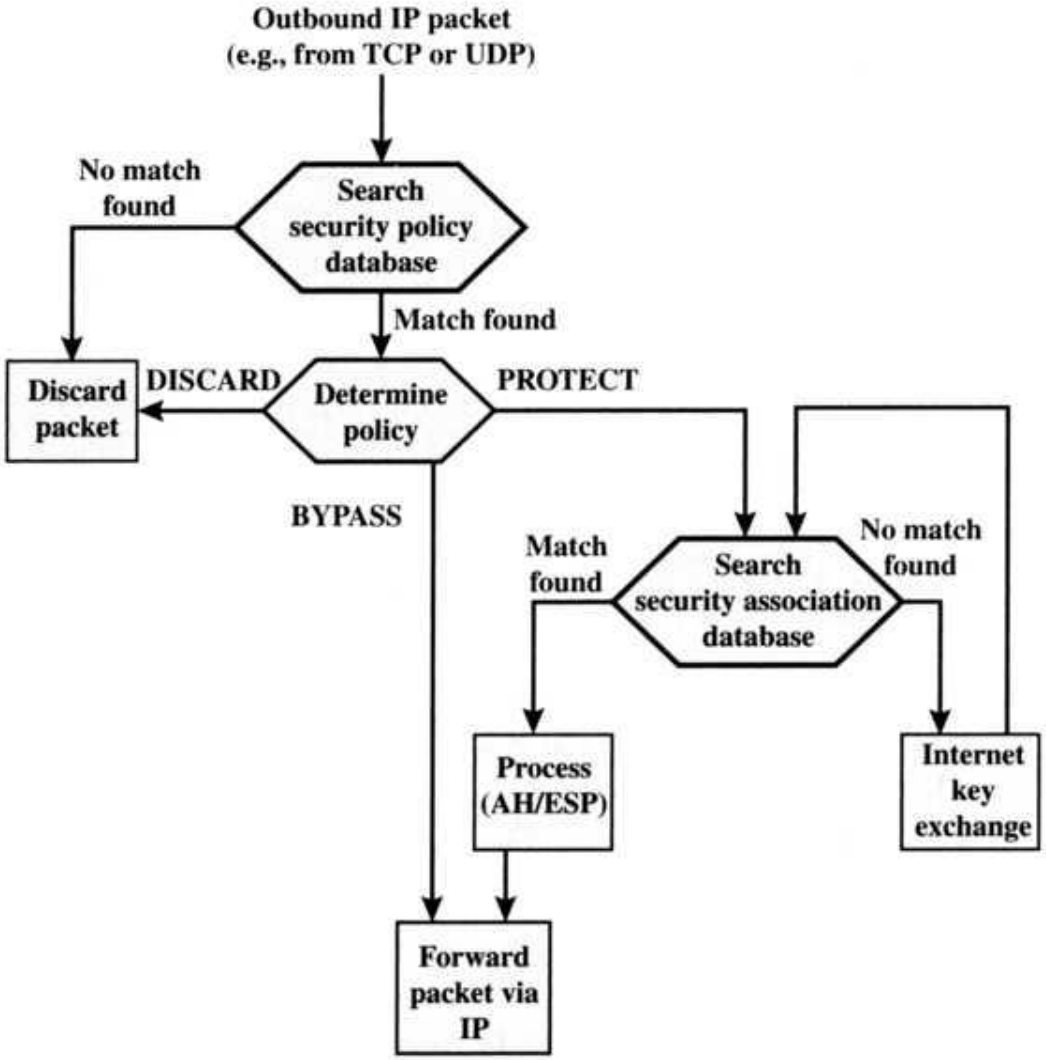
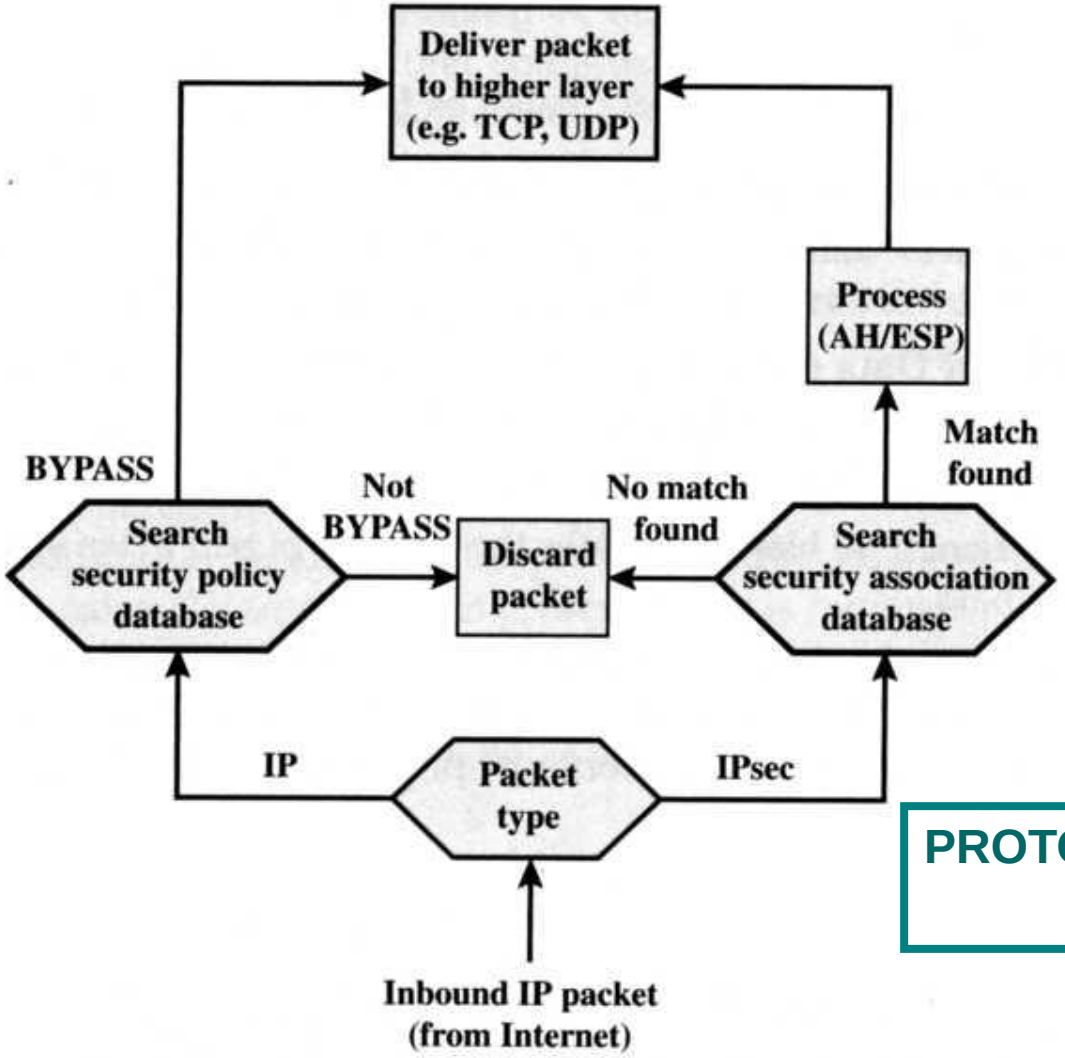


Figure 8.3 Processing Model for Outbound Packets

Note that BYPASS IPsec is a possible policy for non-sensitive traffic that requires no security processing.



PROTOCOL = 50 or 51

Figure 8.4 Processing Model for Inbound Packets

Before a pair of security gateways can exchange user data protected by IPsec, they must complete a preliminary handshake, the ***Internet Key Exchange***.

During the handshake they establish algorithms, keys that will be used, and authenticate each other.

IKE itself has two phases:

- phase 1: a secure channel, the ***IKE Security Association*** pair is set up between the two security gateways

- phase 2: the two gateways use this channel to negotiate safely one or more ***IPsec SA*** pairs that will be used to protect transfer of user data between the two intranets

Both phases must be complete before user data can flow

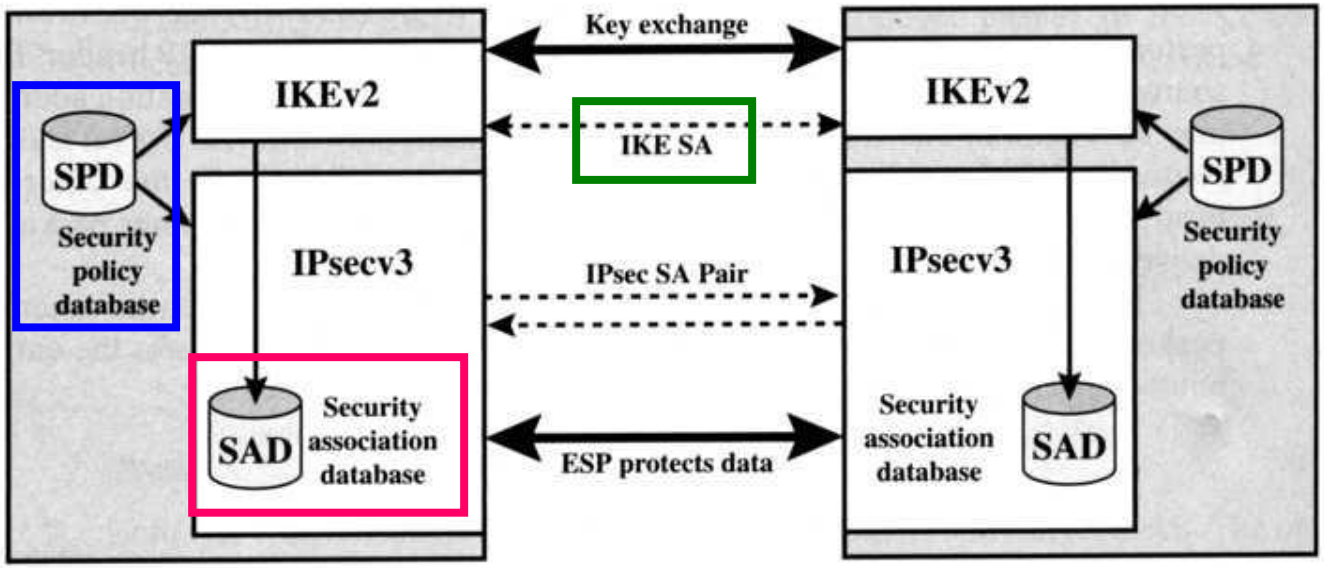


Figure 8.2 IPsec Architecture

First we must establish a secure channel between the two security gateways – the IKE SA

Then the SPD tells IKE what SAs are needed for user traffic.

IKE negotiates the user SAs and enters them in the SAD.

IKE Phases

- IKE has two phases
  - Phase 1: Establish bi-directional IKE SA
    - Note: IKE SA different from IPsec SA
    - Also called ISAKMP security association
  - Phase 2: ISAKMP is used to securely negotiate the IPsec pair of SAs
- Phase 1 has two modes: aggressive mode and main mode
  - Aggressive mode uses fewer messages
  - Main mode provides identity protection and is more flexible

Diffie-Hellman (DH) Key Exchange

Given (g, g^a) hard to compute a – Discrete Logarithm Assumption

1. A → B: $K_a = g^a \bmod p$
2. B → A: $K_b = g^b \bmod p$

3. A outputs $K_{ab} = K_b{}^a$
4. B outputs $K_{ba} = K_a{}^b$

- Note $K_{ab} = K_{ba} = g^{ab} \bmod p$

Security of DH key exchange

- No authentication of either party
- Secure only against a passive adversary
  - Under the computational Diffie-Hellman assumption
    - Given (g, g^a, g^b), hard to compute g^{ab}
- Not secure against an active attacker
  - Man-in-the-middle attack…

Authenticated DH Key Exchange

1. A → B: $K_a = g^a \bmod p$
2. B → A: $Cert_b$, $K_b = g^b \bmod p$
   $Sig_{SKb}(K_b, K_a)$
3. A → B: $Cert_a$, $Sig_{SKa}(K_a, K_b)$

4. A outputs $K_{ab} = K_b^{\,a}$
5. B outputs $K_{ba} = K_a^{\,b}$

IKE phase 1:

The ***Identity Protection Exchange*** consists of 6 messages:

Messages (1) and (2): Peers negotiate algorithms to be used for establishment of the secure channel between security gateways

Messages (3) and (4) generate the keys to be used

Messages (5) and (6) authenticate the peers.

IKE Phase 2

We could go through the six-message exchange again, selecting algorithms, producing totally new keys and checking authentication.

Instead, use ***Quick Mode***, which takes just three messages.

Quick Mode accepts the choice of algorithms made in phase1.

It also accepts the DH key generated in phase 1, but hashes it to make “new” keys (“new keys from old”)

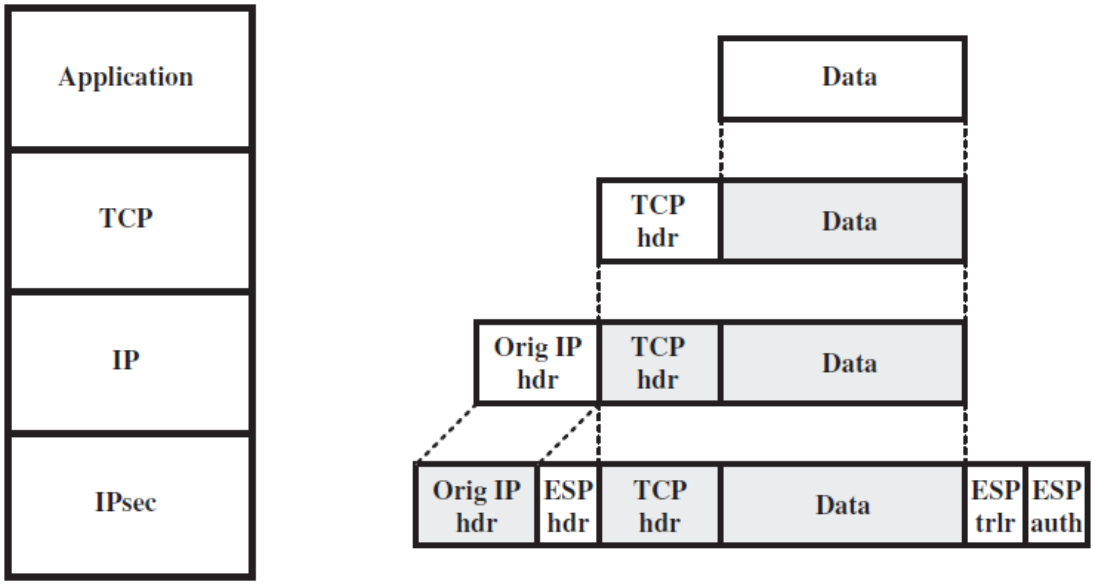
End of Handshake.

IKE and IPSec Security Associations established.

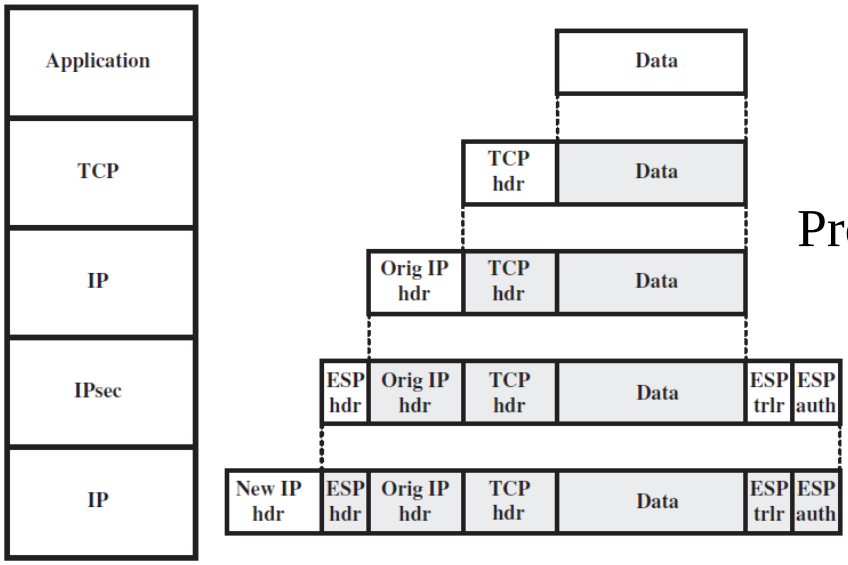
We can proceed to transmit user data.

Summary of IPsec

- IKE message exchange for algorithms, secret keys, SPI numbers
- Either the AH or the ESP protocol (or both)
- The AH protocol provides integrity and source authentication
- The ESP protocol (with AH) additionally provides encryption
- IPsec peers can be two end systems, two routers/firewalls, or a router/firewall and an end system



(a) Transport mode



(b) Tunnel mode

Protocol Operation for ESP