IP AND IPSEC

EN.600.424

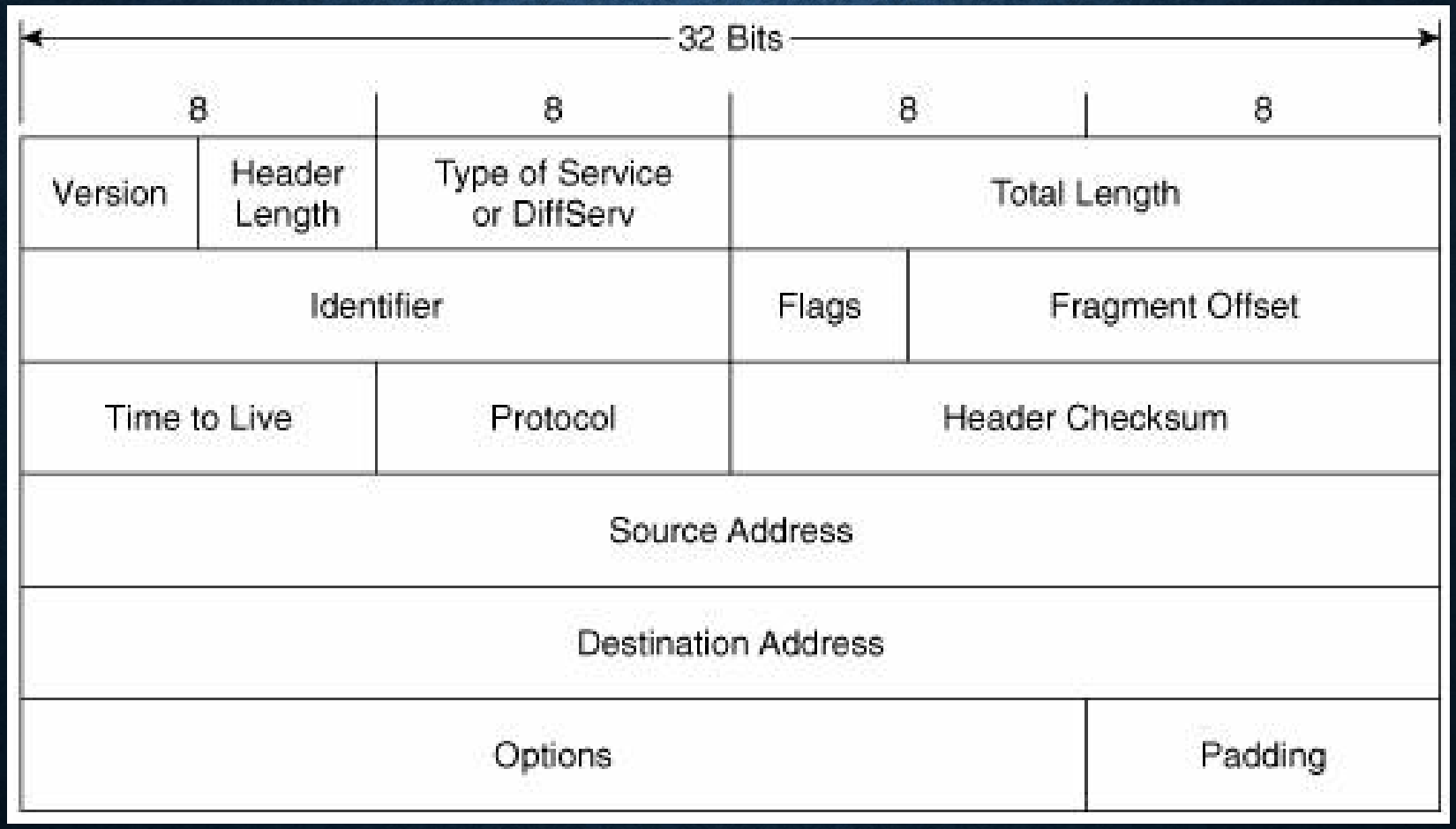
Fall 2018

Lecture Notes

IPV4 PROTOCOL

- Layer 3 Protocol
- Handles fragmentation and reassembly
  - Assumed that across multiple LANS, multiple MAC protocols
  - Each MAC protocol might have its own MTU
- Also, of course, includes the global IP address
  - Kind of global…

IPV4 HEADER


32 Bits
8
8
8
8
Version
Header Length
Type of Service or DiffServ
Total Length
Identifier
Flags
Fragment Offset
Time to Live
Protocol
Header Checksum
Source Address
Destination Address
Options
Padding

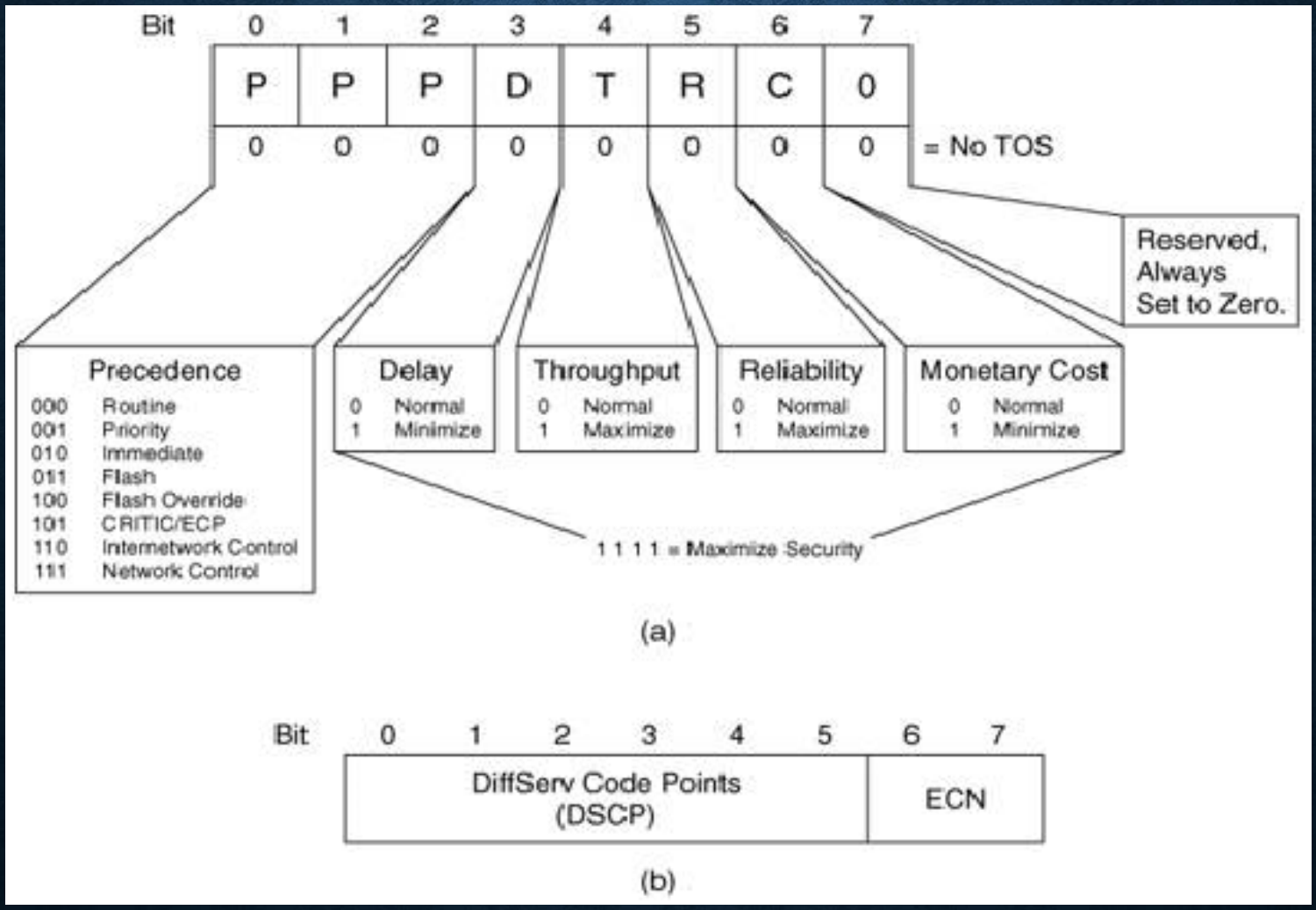
IPV4 HEADER FIELDS

- Version – 0100 (binary 4)
- Header Length – Length of header in 4-byte increments
- Total Length – Size of header and data in bytes (max 65535)
- Identification – for recognizing fragments
- Flags
  - Reserved. Always 0
  - Don't fragment
  - More fragments

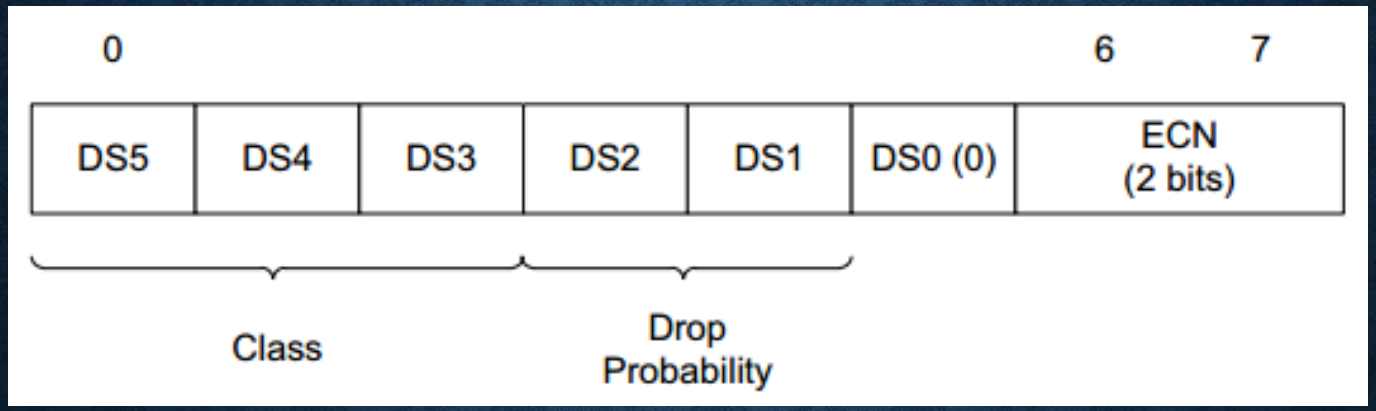
MORE IPV4 HEADER FIELDS

- Fragment Offset – Offset in original datagram
- TTL – Counter to prevent infinite routing
- Protocol – Information about upper layer (17=UDP, 6=TCP)
- Header Checksum
  - Recomputed each hop (because of TTL changes)
  - Checksum field itself always presumed to be 0

ORIGINAL TOS



DIFFERENTIATED SERVICES



ECN

- Explicit Congestion Notification
- When enabled, indicates congestion without dropping
- Not all hardware/software supports ECN

IPV4 ADDRESSES

- IPv4 Addresses are a.b.c.d where each is between 0-255
- In actuality, just a 32-bit number ("four octets")
  - 192.0.2.235
  - 3221226219
  - 0xC00002EB (0xC0.0x00.0x02.0xEB)
- Private Networks:
  - 10.0.0.0
  - 172.16.0.0
  - 192.168.0.0

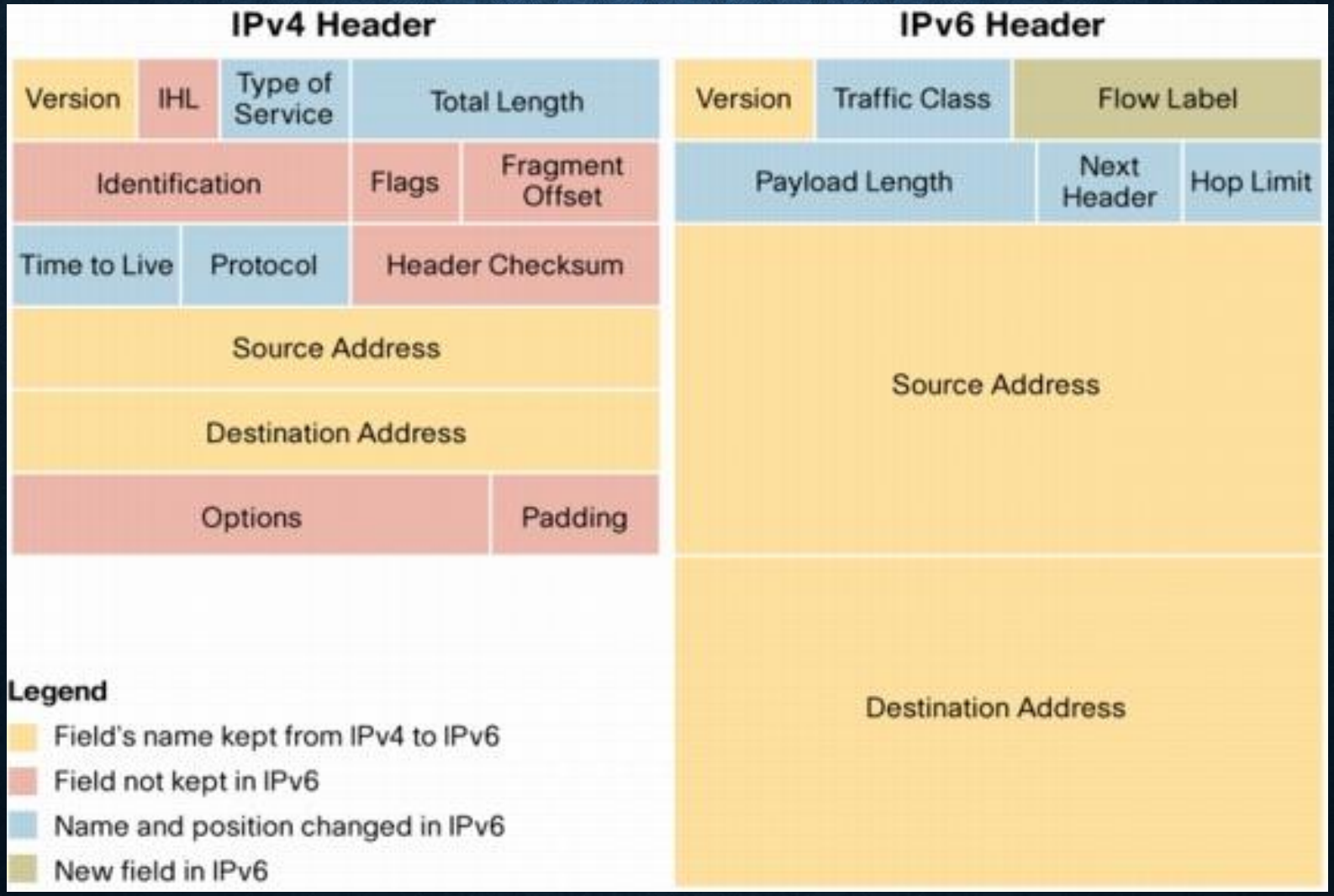
FRAGMENTATION

- If packet size > MTU, and DNF is 0
  - Each fragment gets its own size
  - Each fragment except the last gets MF set to 1
  - Fragment offset is location in the original packet
  - Identification field is unique identifier of original datagram
- Reassembly
  - Use src, dst, protocol, and identification to identify fragments
  - Use offset to store data in reassembly buffer
  - Use MF = 0 to recognize end of reassembly

FRAGMENTATION ISSUES

- IPv4 Fragmentation had security issues
  - IP Fragmentation Overlap (overwrite a fragment)
  - Buffer full (too many incomplete fragments)
  - Fragment overrun
  - (Note that most of these are DoS, but some evasion)

IPV6

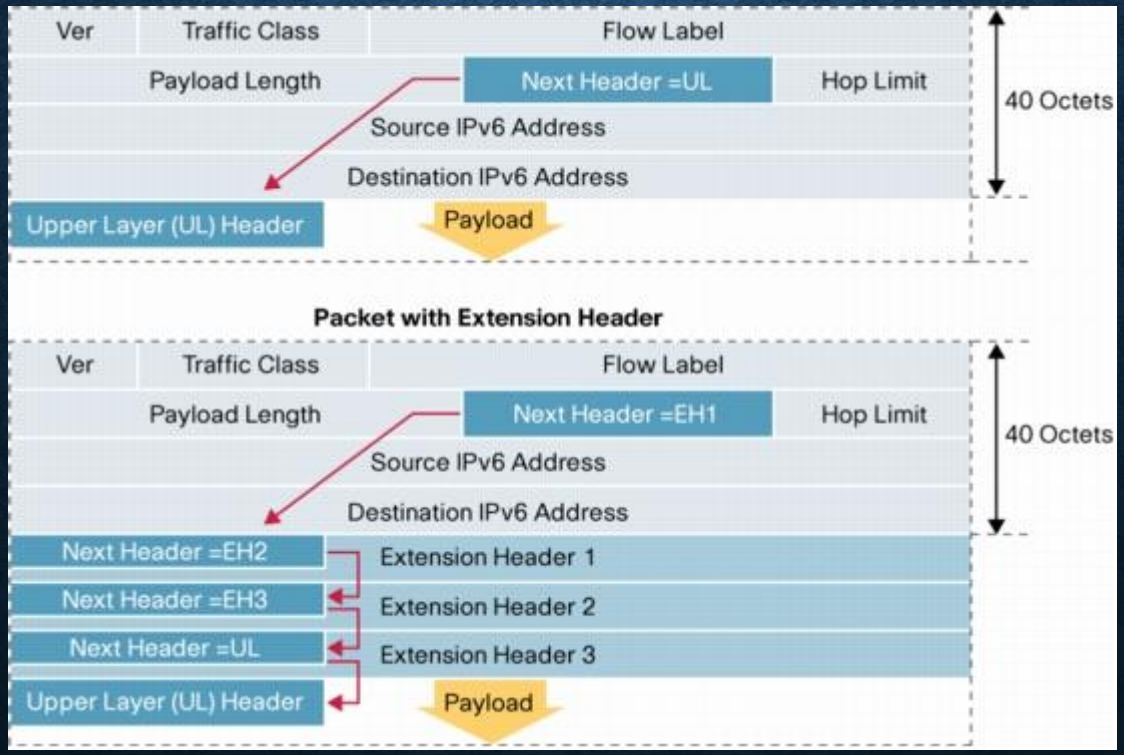

IPv4 Header
Version
IHL
Type of Service
Total Length
Identification
Flags
Fragment Offset
Time to Live
Protocol
Header Checksum
Source Address
Destination Address
Options
Padding
IPv6 Header
Version
Traffic Class
Flow Label
Payload Length
Next Header
Hop Limit
Source Address
Destination Address
Legend
Field's name kept from IPv4 to IPv6
Field not kept in IPv6
Name and position changed in IPv6
New field in IPv6

IPV6 HEADER FIELDS

- Version = 0110 (binary 6)
- Traffic Class: Differentiated services plus ECN
- Flow label: Hint for multiple outbound paths
- Payload length: Includes extension headers, in bytes
- Next header: Type of next extension or transport header
- Hop limit: Decrement by 1, discard if 0

EXTENSION HEADERS

- IPv6 is always 40 bytes for main header
- Can have additional headers:



COMMON EXTENSION HEADERS

Order	Header Type	Next Header Code
1	**Basic IPv6 Header**	-
2	**Hop-by-Hop Options**	0
3	**Destination Options (with Routing Options)**	60
4	**Routing Header**	43
5	**Fragment Header**	44
6	**Authentication Header**	51
7	**Encapsulation Security Payload Header**	50
8	**Destination Options**	60
9	**Mobility Header**	135

IPV6 FRAGMENTATION

- To deal with IPv4 Frag issues, ONLY SENDER can frag in IPv6
- Ergo, ***sender must know smallest MTU of path!***
- Path MTU Discovery (PMTUD)
  - If too big, send an ICMPv6 "packet too big" to sender
- Otherwise, max IPv6 packet size is 1,280 bytes.
- Uses a fragmentation extension header
  - Identification
  - MF, etc

IPV6 FRAG PROBLEMS

- Studies from 2014-present indicate IPv6 fragmentation fails
- About one-third of IPv6 hosts could not receive frags
- Many are concluding that IPv6 fragmentation is deprecated
- Maximum IPv6 packet size between 1280 and 1350
- See,
  - https://blog.apnic.net/2016/05/19/fragmenting-ipv6/
  - https://labs.apnic.net/?p=1033

IPSEC

- IPSec provides the following protocols:
  - Authentication Headers – integrity of immutable fields
  - Encapsulating Security Payloads – confidentiality
  - Security Associations – negotiate parameters of AH, ESP
    - Internet Key Exchange (IKE)

AH FIELDS

- *Operates **ON TOP** of IP*!
  - Protocol Number 51
- Next Header (e.g., TCP)
- Payload Length – 4-octet units minus 2
- Security Parameters Index – With dest address forms SA
- Sequence Number – increasing number to prevent replay
- Integrity Check Value – the authenticated hash

AH VISUAL

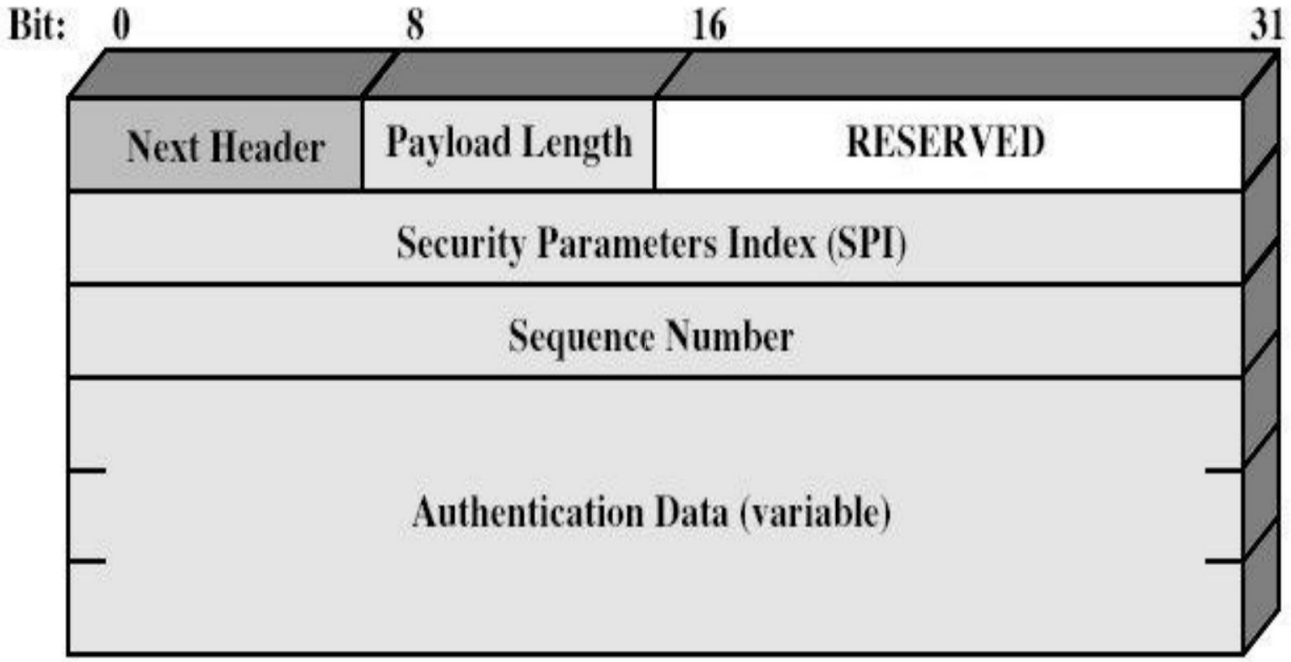
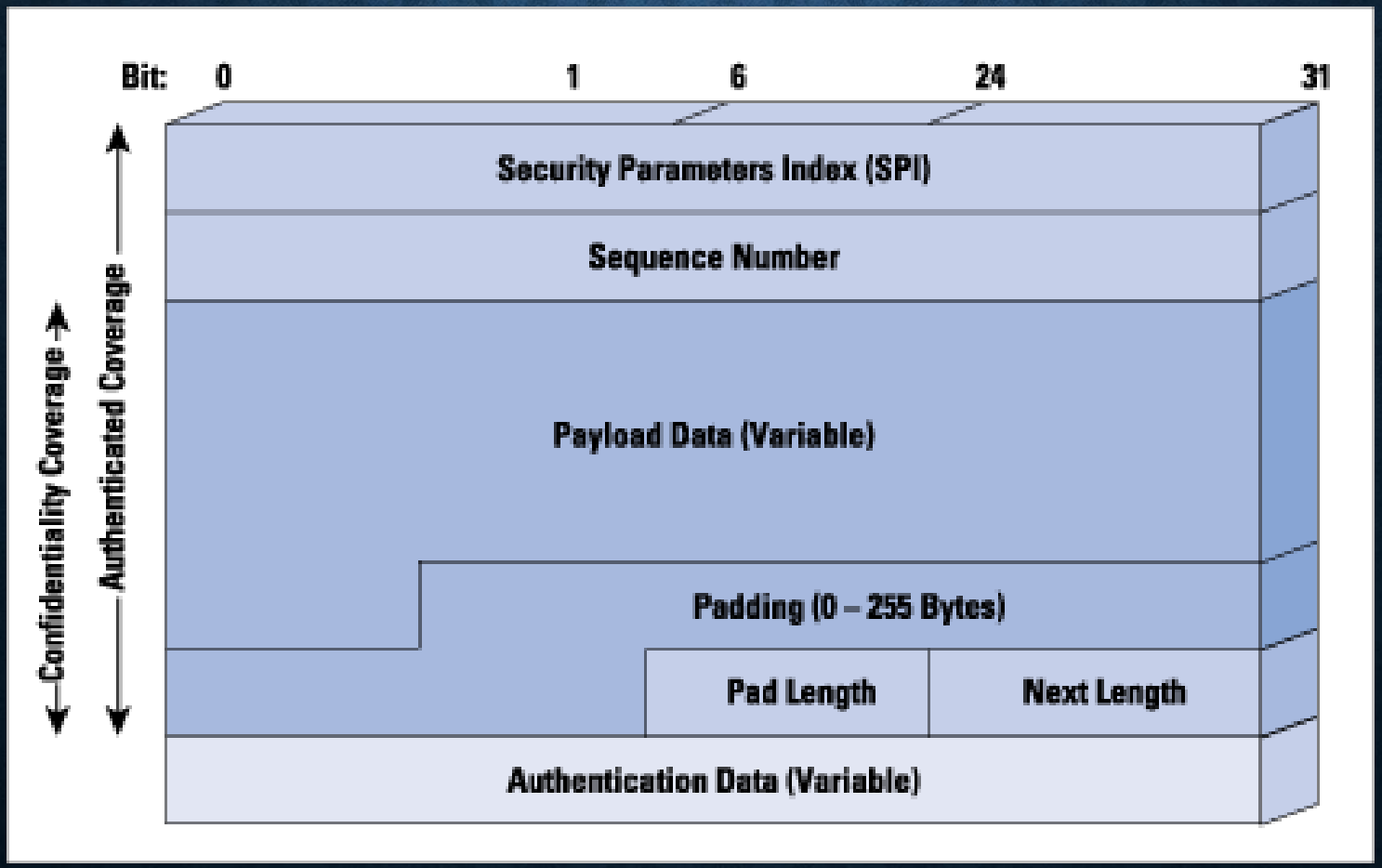


Figure 1.3 IPSec Authentication Header

ESP FIELDS

- Operates on top of IP, protocol number 50
- Security Parameter Index – With dest addr, identifies the SA
- Sequence Number – Increasing number to prevent replay
- Protected content of the original IP packet
- Padding for encryption
- Pad Length – Size of pad in octets
- Next Header (e.g., TCP)
- Integrity Check Value – authenticated hash

ESP VISUAL


Bit:
0
1
6
24
31
Security Parameters Index (SPI)
Sequence Number
Payload Data (Variable)
Padding (0 – 255 Bytes)
Pad Length
Next Length
Authentication Data (Variable)
Confidentiality Coverage
Authenticated Coverage

IKE

- Internet Key Exchange
- IKEv1 (RFC 2409 + many updates)
- IKEv2 (RFC 4306 + many updates)
  - Most recent version is 7296, updated by 7427, 7670, 8247

IKEV1

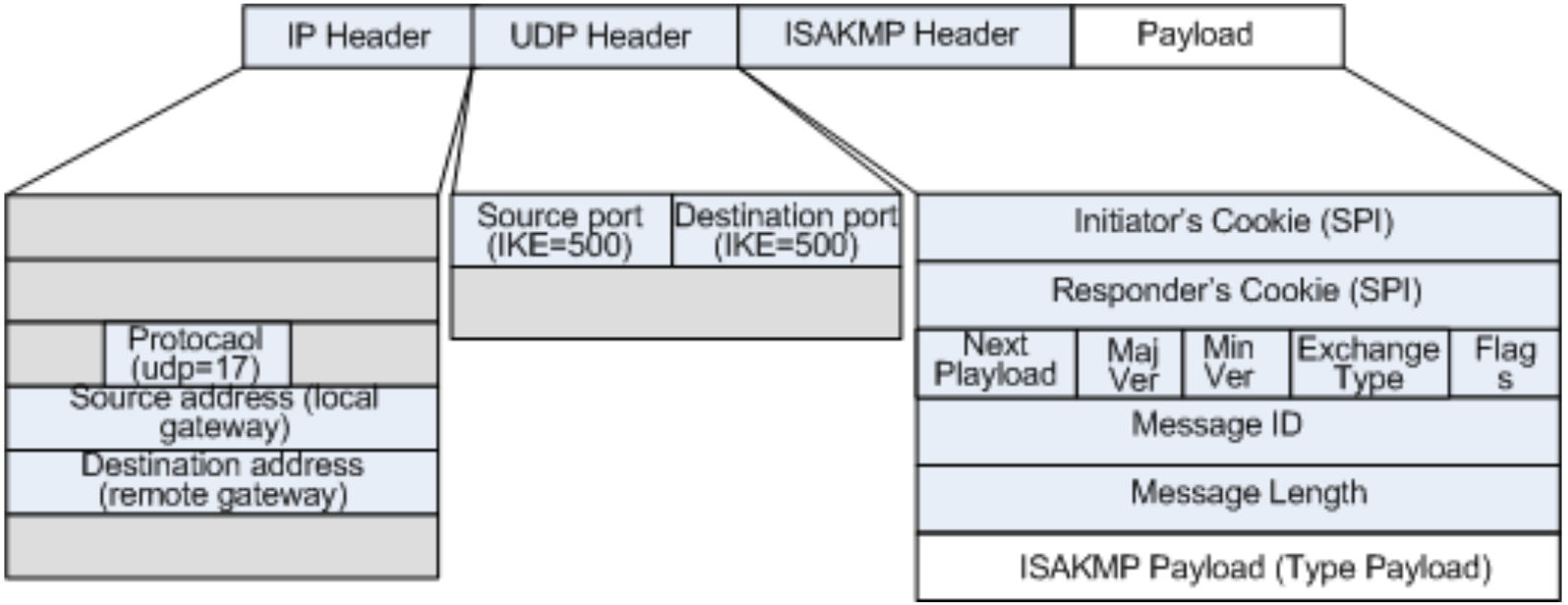
- RFC 2409
- Contains three protocols:
  - ISAKMP provides a framework for authentication and key exchange but does not define them
  - Oakley describes a series of key exchanges
  - SKEME describes a versatile key exchange technique which provides anonymity, repudiability, and quick key refreshment.
- Uses part of Oakley and part of SKEME in conjunction with ISAKMP to obtain authenticated keying material

IKEV1 PROTOCOL PHASES

- Phase 1 - where the two ISAKMP peers establish a secure, authenticated channel with which to communicate. This is called the ISAKMP Security Association (SA)
- Phase 2 - Phase 2 is where Security Associations are negotiated on behalf of services such as IPsec or any other service which needs key material and/or parameter negotiation.
- One phase 1 can permit many phase 2's

ISEKMP

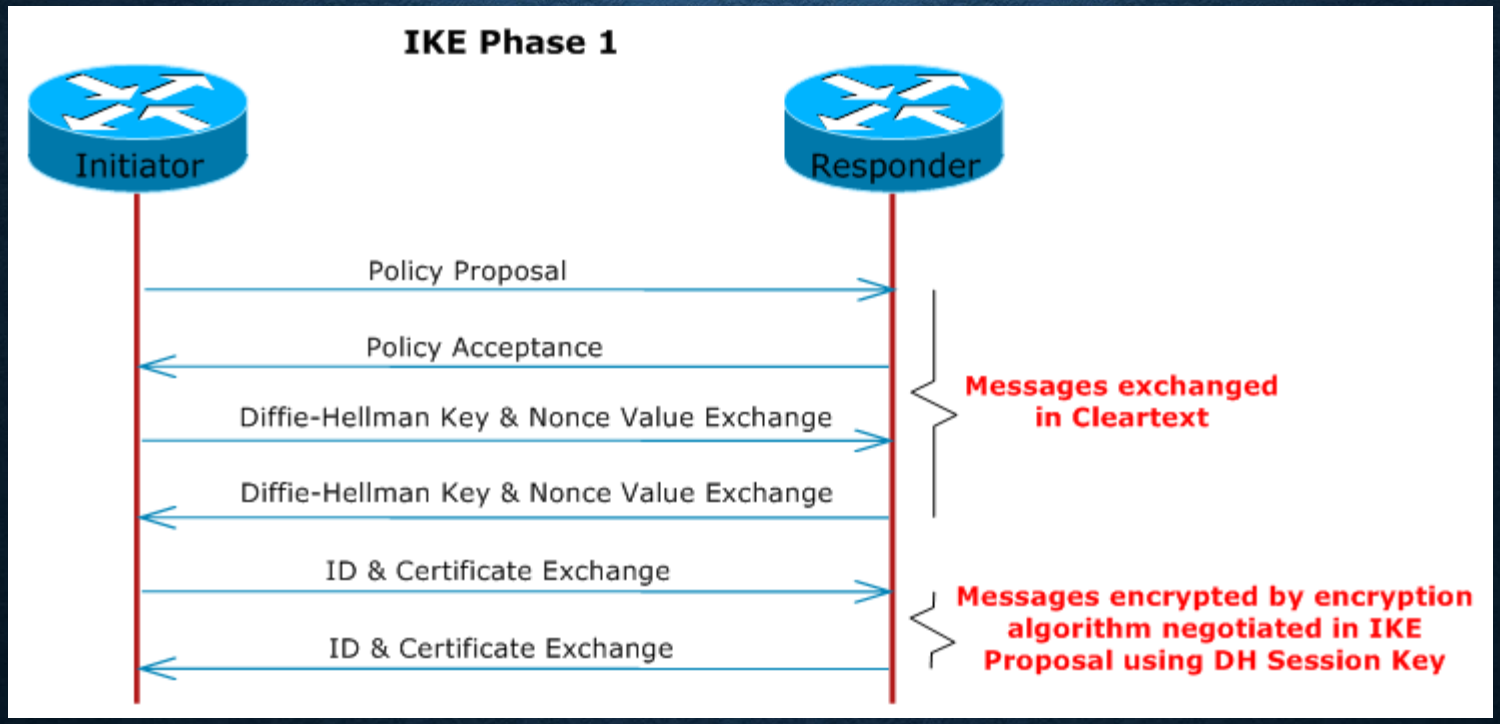
Figure 1-1 ISAKMP packet encapsulation and packet headers



ISAKMP HEADER

- Initiator Cookie – Cookie of initiating device
- Responder's Cookie – Must be 0 on first message!
- Next Payload – Indicates ISAKMP payload (examples:)
  - Security Association
  - Proposal
  - Transform
- Exchange Type (phase 1, phase 2)

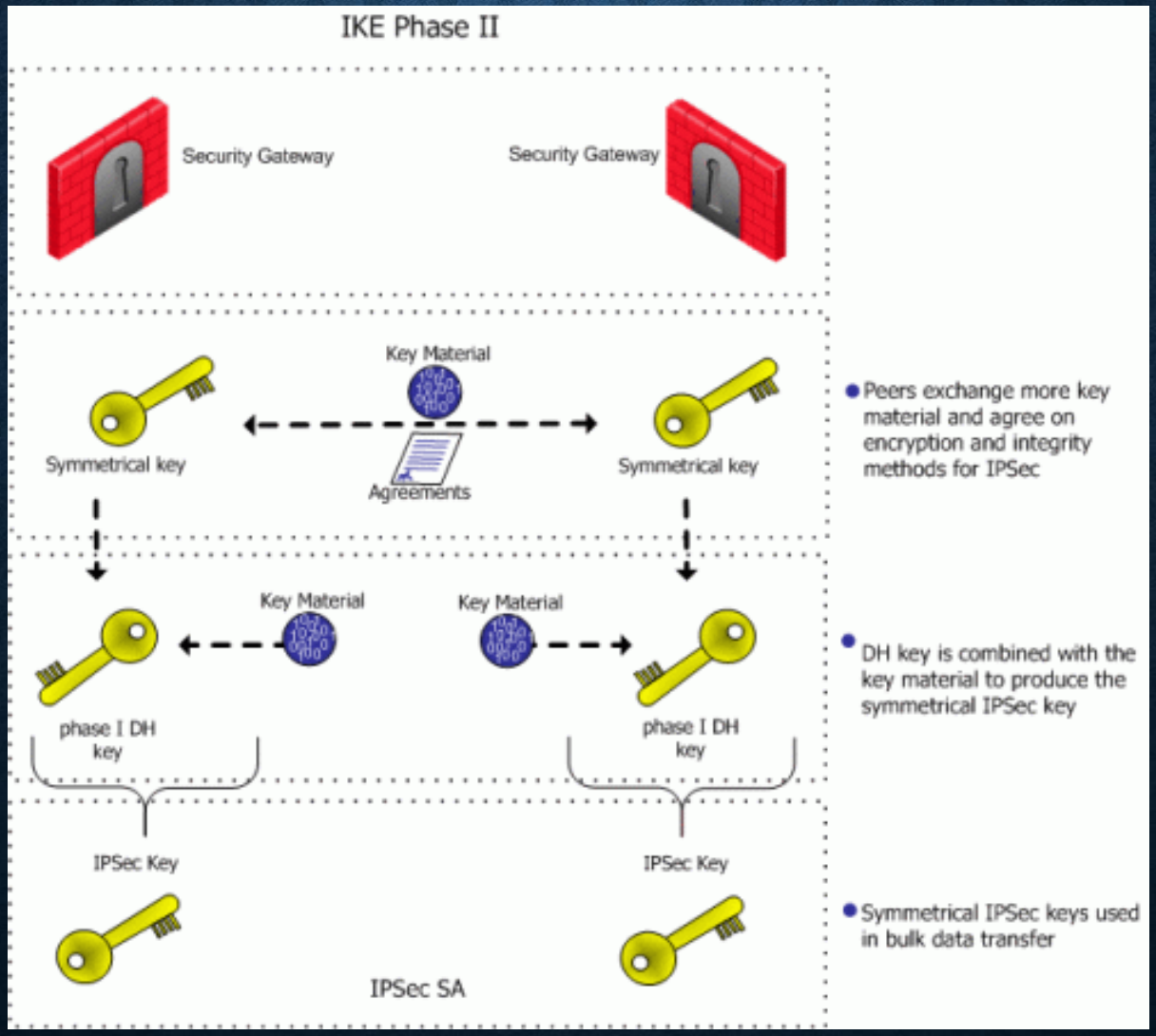
ISAKMP EXCHANGE



PHASE 1 AUTHENTICATION

- Pre-shared keys
- Digital Signatures
- Public-key *Encryption*
- Revised mode, Public-key Encryption

PHASE 2


IKE Phase II
Security Gateway
Security Gateway
Key Material
Symmetrical key
Agreements
Symmetrical key
Peers exchange more key material and agree on encryption and integrity methods for IPSec
Key Material
Key Material
phase I DH key
phase I DH key
DH key is combined with the key material to produce the symmetrical IPSec key
IPSec Key
IPSec Key
IPSec SA
Symmetrical IPSec keys used in bulk data transfer

IKEV2

- Significantly less complicated than IKEv1
- IKEv1 sends at least 6-9 messages for setup
  - 6 if using "aggressive" mode
  - 9 if using "main" mode
- IKEv2 sends 4 messages total
  - SA exchange
  - AUTH exchange

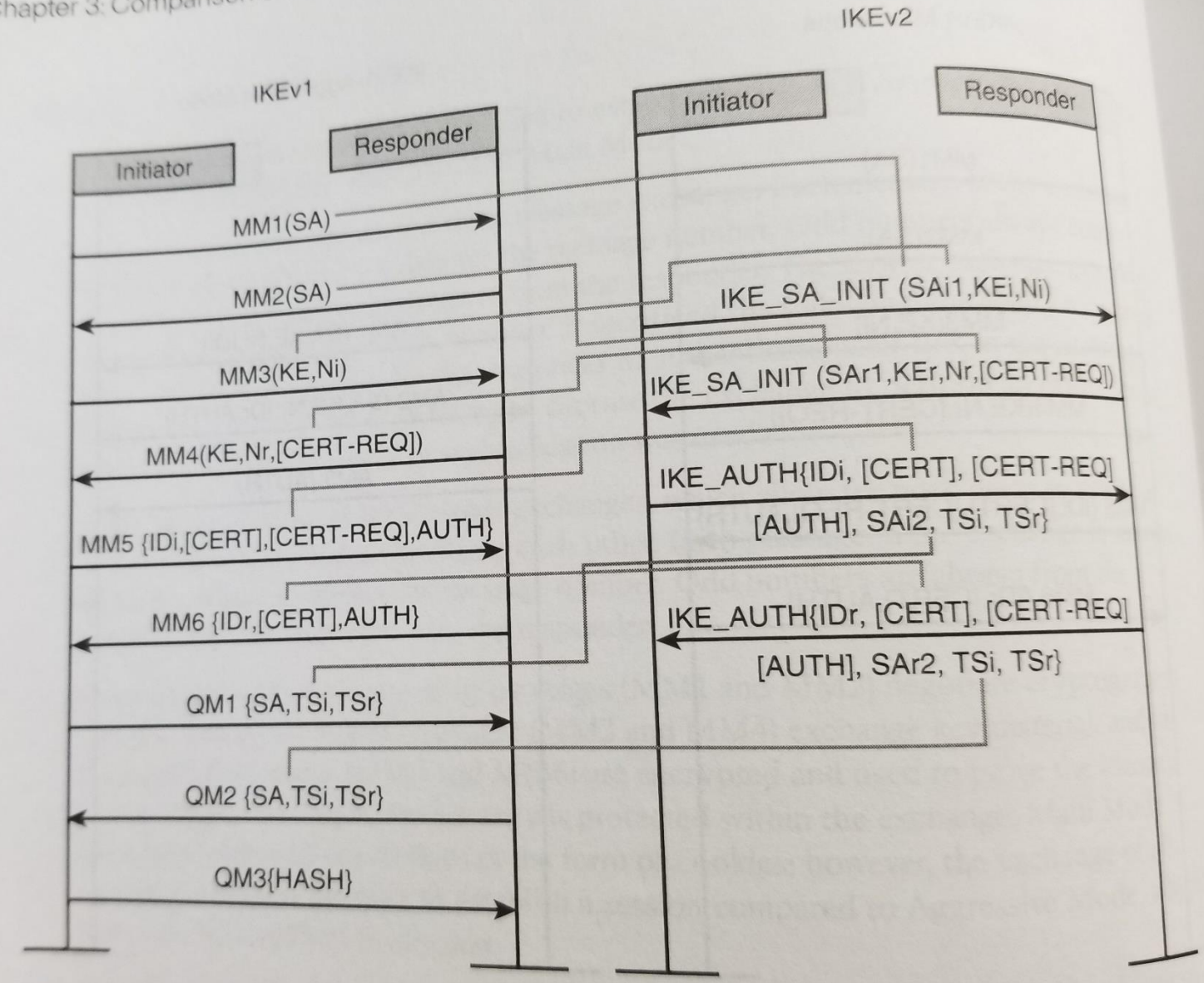


Figure 3-2 *Relationship of Attributes Sent in IKEv1 Exchanges, Compared to IKEv2*

If additional IPsec S

IKEV2 AUTH

- Pre-shared Keys
- Digital Signatures
- EAP (Extensible Access Protocol)
  - ONLY FOR VALIDATING THE INITIATOR
  - Can
- Asymmetric Authentication

identity; this allows for separate IKE and EAP identities to exist for a device.

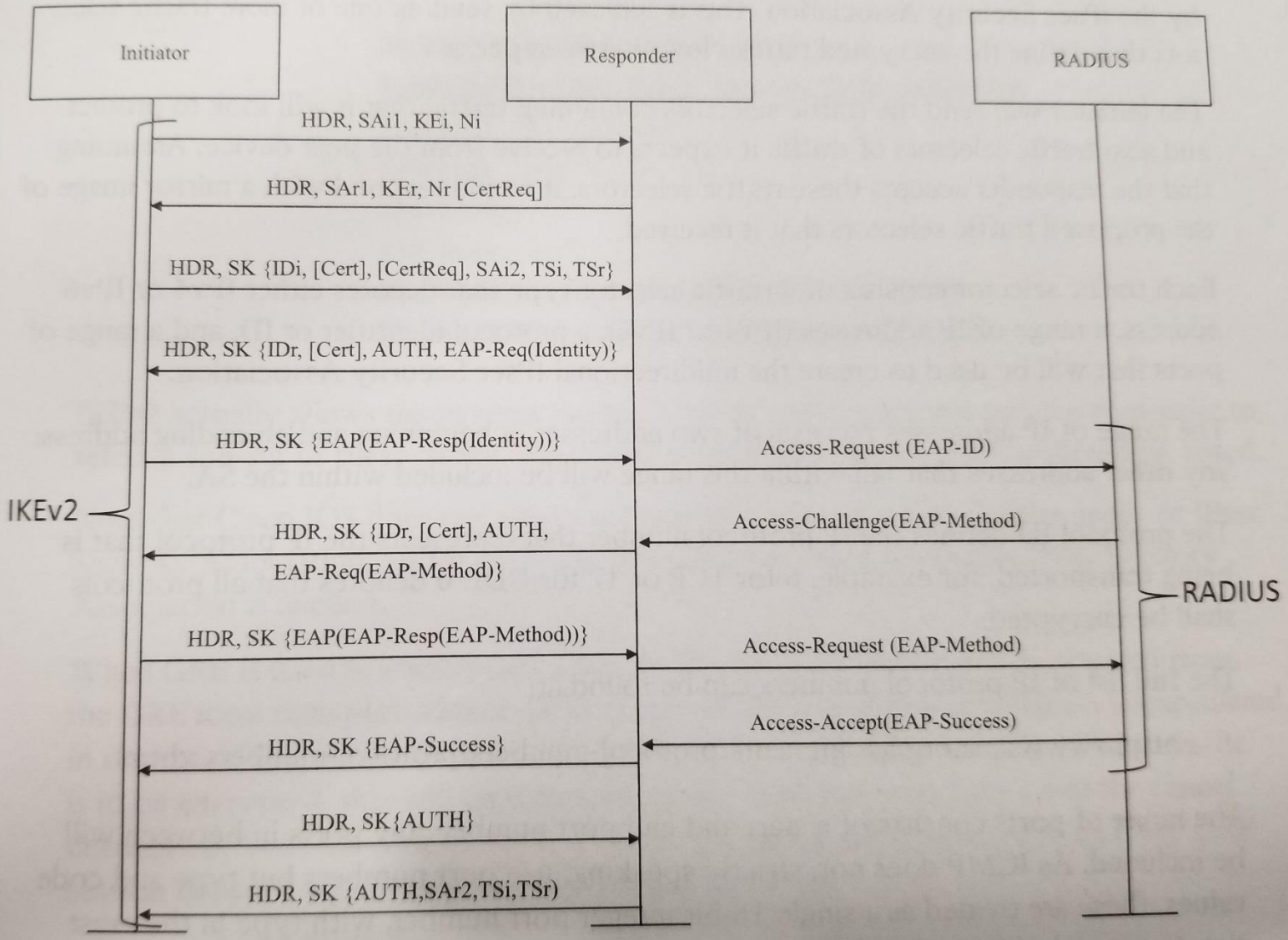


Figure 2-8 *EAP with Query-Identity Used*

ADDITIONAL IKE2 BENEFITS

- High Availability
- Identities + Parallel SA's between peers
- "Built-in" NAT
- Mobility and Multi homing
- Request-response (retry) reliability
- Combined-mode Ciphers

MODES OF OPERATION

- Transport Mode
  - AH or ESP on payload of IP Packet
- Tunnel Mode
  - Entire IP Packet is encrypted and/or authenticated
  - Put into a NEW IP packet (i.e., VPN tunnel)
- Can use AH and ESP, but typically shouldn't.