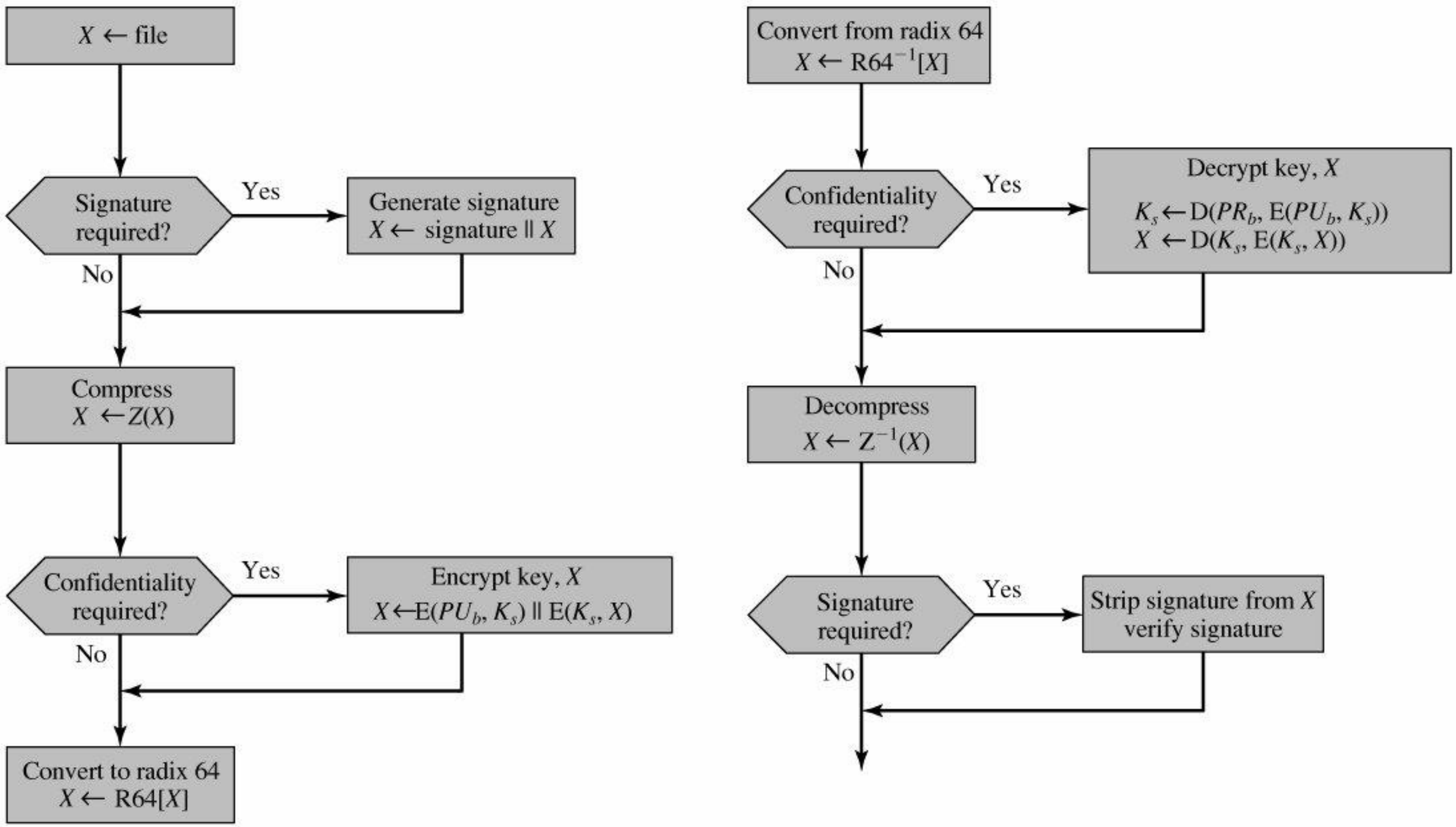
IT 422 Network Security
IP Security

Yasser F. O. Mohammad

REMINDER 1: How can Email be enhanced

- confidentiality
  - protection from disclosure
- authentication
  - of sender of message
- message integrity
  - protection from modification
- non-repudiation of origin
  - protection from denial by sender

REMINDER 2: Transmission and Reception



(a) Generic transmission diagram (from A)

(b) Generic reception diagram (to B)

REMINDER 3: Key Ring

Private-Key Ring

Timestamp	Key ID*	Public Key	Encrypted Private Key	User ID*
⋮	⋮	⋮	⋮	⋮
T_i	$PU_i \bmod 2^{64}$	PU_i	$E(H(P_i), PR_i)$	User i
⋮	⋮	⋮	⋮	⋮

Public-Key Ring

Timestamp	Key ID*	Public Key	Owner Trust	User ID*	Key Legitimacy	Signature(s)	Signature Trust(s)
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
T_i	$PU_i \bmod 2^{64}$	PU_i	trust_flag$_i$	User i	trust_flag$_i$		
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

* = field used to index table

REMINDER 4: Functions of S/MIME

- Enveloped Data
  - Confidentiality
- Signed Data
  - Authentication
- Clear-signed Data
  - Authentication (RADIX64 applied to signature only for readability)
- Signed and Enveloped Data
  - Confidentiality and Authentication

REMINDER 5: EnvelopedData

1. Generate a pseudorandom session key for a particular symmetric encryption algorithm (RC2/40 or tripleDES).

2. For each recipient, encrypt the session key with the recipient's public RSA key.

3. For each recipient, prepare a block known as RecipientInfo that contains an identifier of the recipient's public-key certificate,[3] an identifier of the algorithm used to encrypt the session key, and the encrypted session key.

4. Encrypt the message content with the session key.

REMINDER 6: SignedData

- Select a message digest algorithm (SHA or MD5).
- Compute the message digest, or hash function, of the content to be signed.
- Encrypt the message digest with the signer's private key.
- Prepare a block known as SignerInfo that contains the signer's public-key certificate, an identifier of the message digest algorithm, an identifier of the algorithm used to encrypt the message digest, and the encrypted message digest.

IP Security

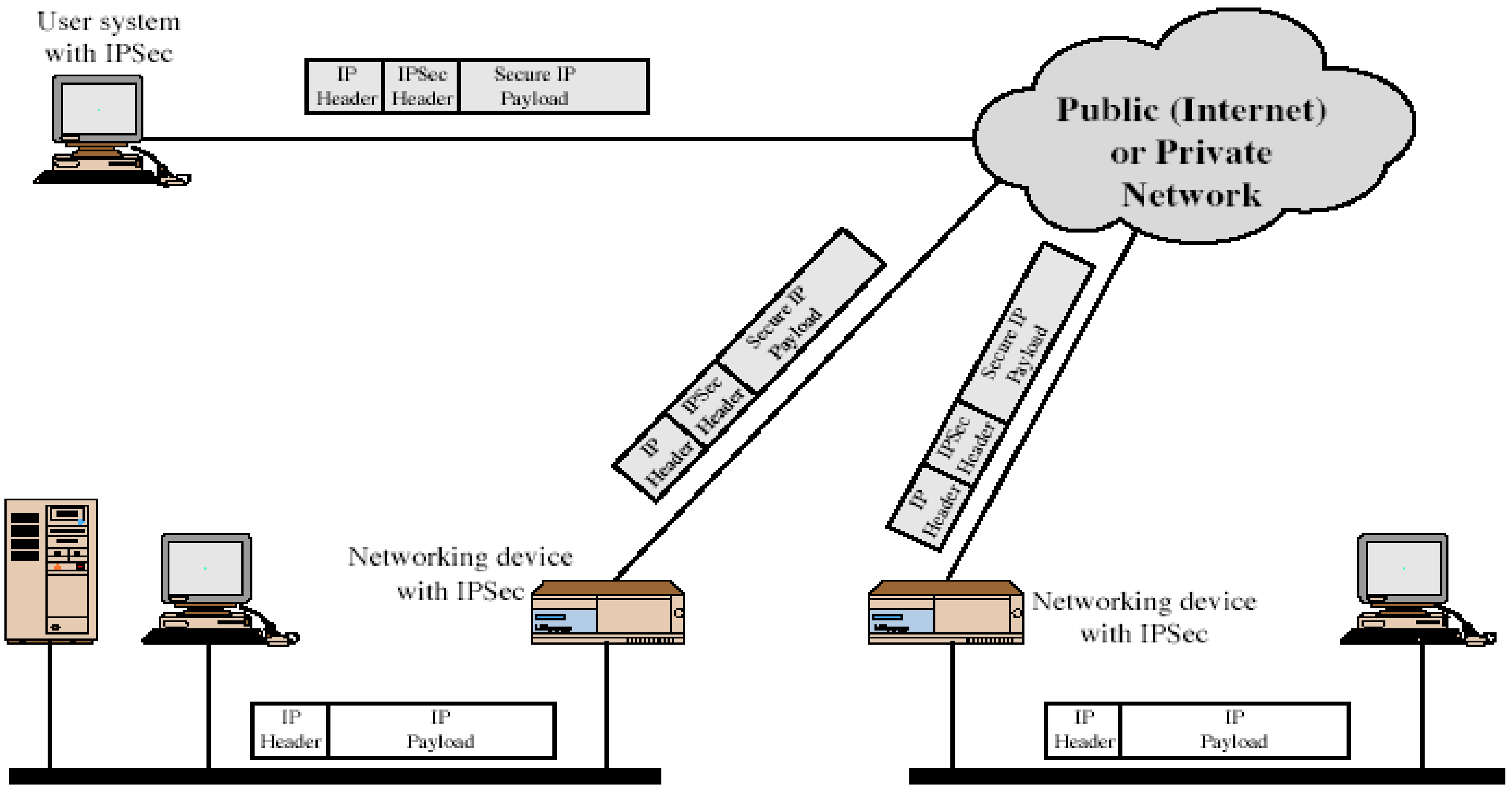
- Applies security services to ALL traffic
- Link encryption
- Useful :
  - No need to modify old applications
  - No need to train employees

IPSec

- General IP Security mechanisms
- Provides
  - authentication
  - confidentiality
  - key management
- applicable to use over LANs, across public & private WANs, & for the Internet

- Available for IPv4 (optional) and IPv6 (required)

IPSec Uses



User system
with IPSec
IP Header
IPSec Header
Secure IP Payload
Public (Internet) or Private Network
IP Header
IPSec Header
Secure IP Payload
IP Header
IPSec Header
Secure IP Payload
Networking device
with IPSec
Networking device
with IPSec
IP Header
IP Payload
IP Header
IP Payload

Benefits of IPSec

- in a firewall/router provides strong security to all traffic crossing the perimeter
- is resistant to bypass
- is below transport layer, hence transparent to applications
- can be transparent to end users

IP Security Architecture

- specification is quite complex
- defined in numerous RFC's
  - incl. RFC 2401/2402/2406/2408
  - many others, grouped by category
- mandatory in IPv6, optional in IPv4

IPSec Services

- Access control
- Connectionless integrity
- Data origin authentication
- Rejection of replayed packets
  - a form of partial sequence integrity
- Confidentiality (encryption)
- Limited traffic flow confidentiality

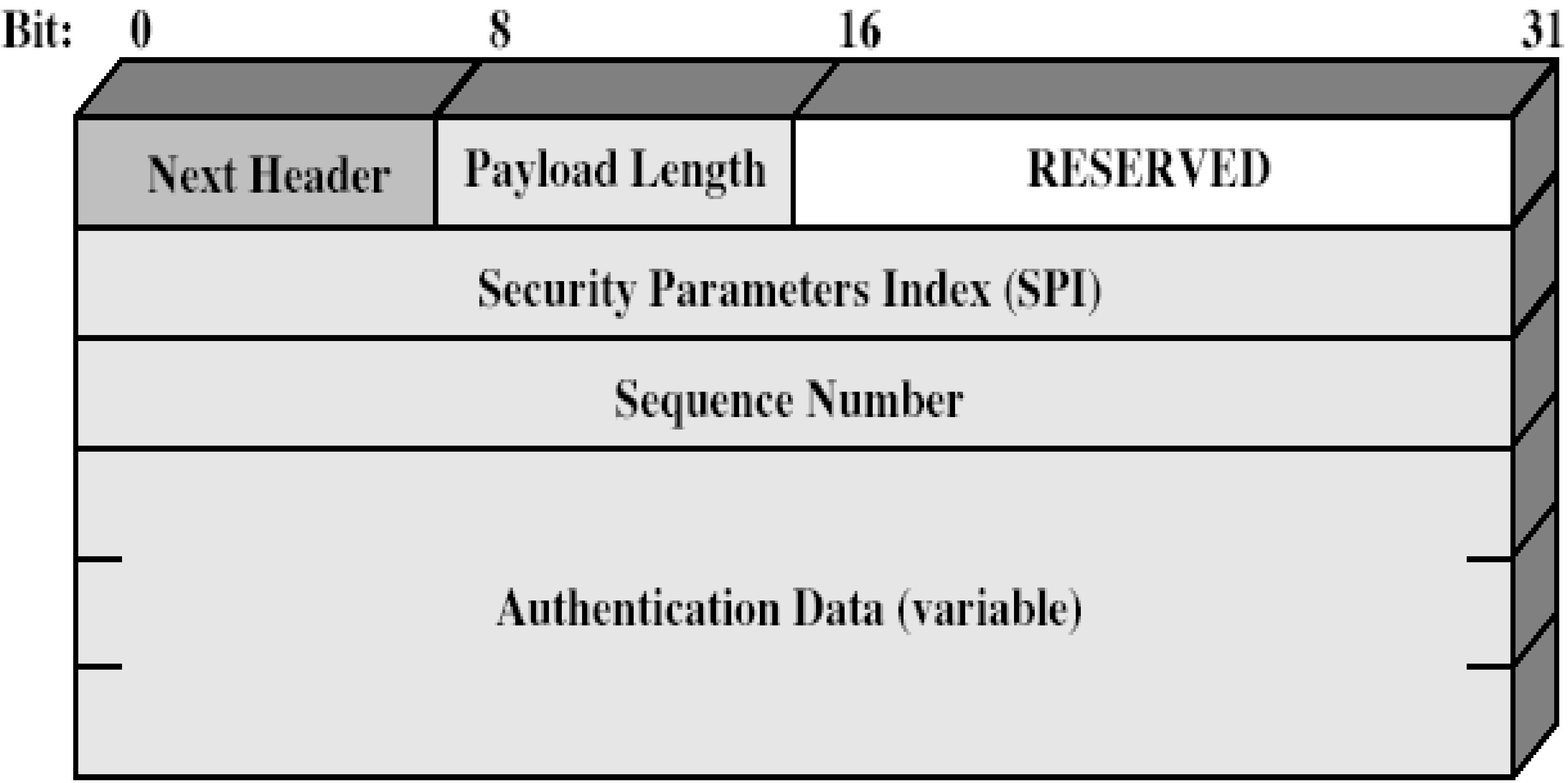
Security Associations

- a one-way relationship between sender & receiver that affords security for traffic flow
- defined by 3 parameters:
  - Security Parameters Index (SPI)
  - IP Destination Address
  - Security Protocol Identifier
- has a number of other parameters
  - seq no, AH & EH info, lifetime etc
- have a database of Security Associations

Authentication Header (AH)

- provides support for data integrity & authentication of IP packets
  - end system/router can authenticate user/app
  - prevents address spoofing attacks by tracking sequence numbers
- based on use of a MAC
  - HMAC-MD5-96 or HMAC-SHA-1-96
- parties must share a secret key

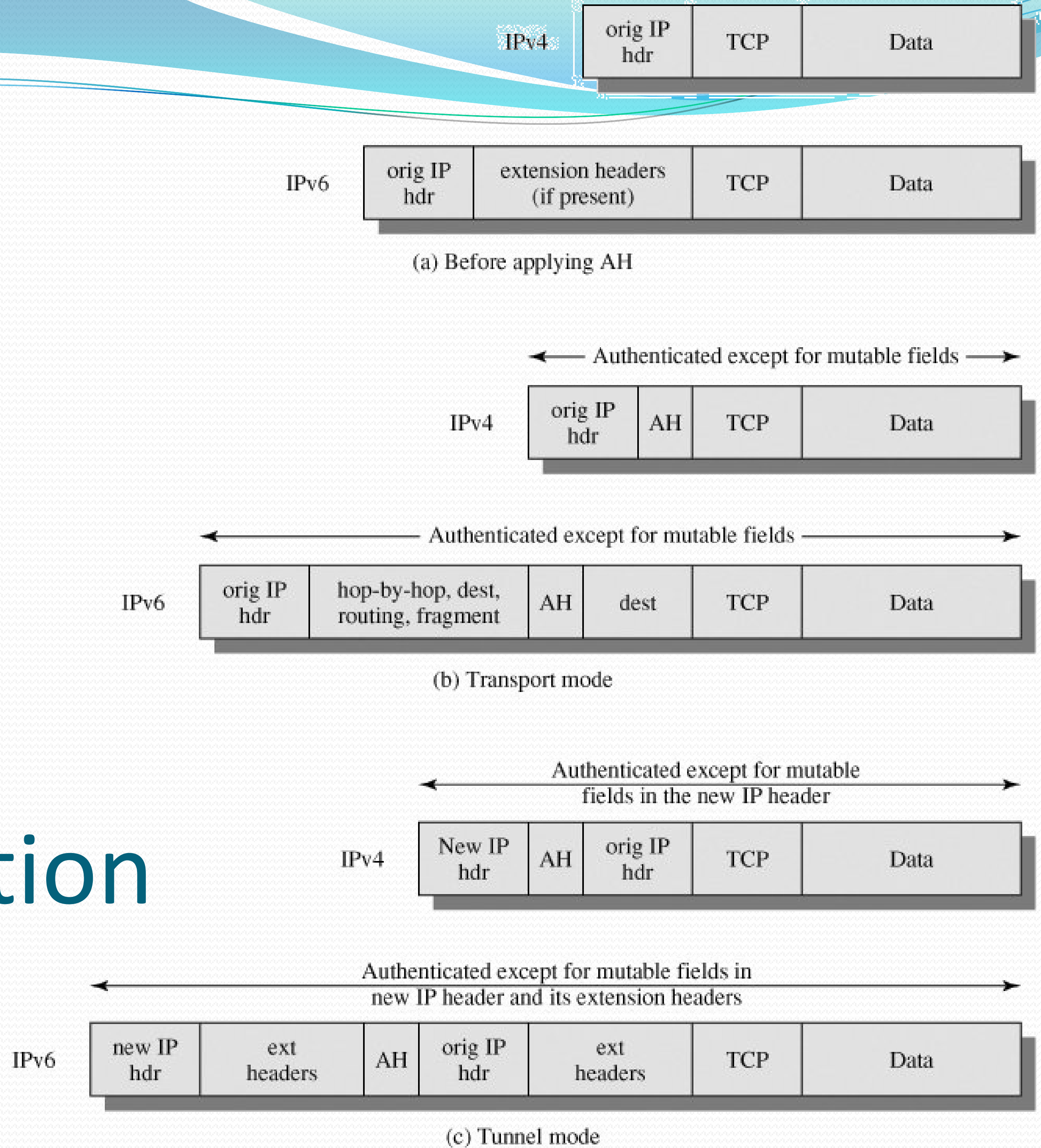
Authentication Header



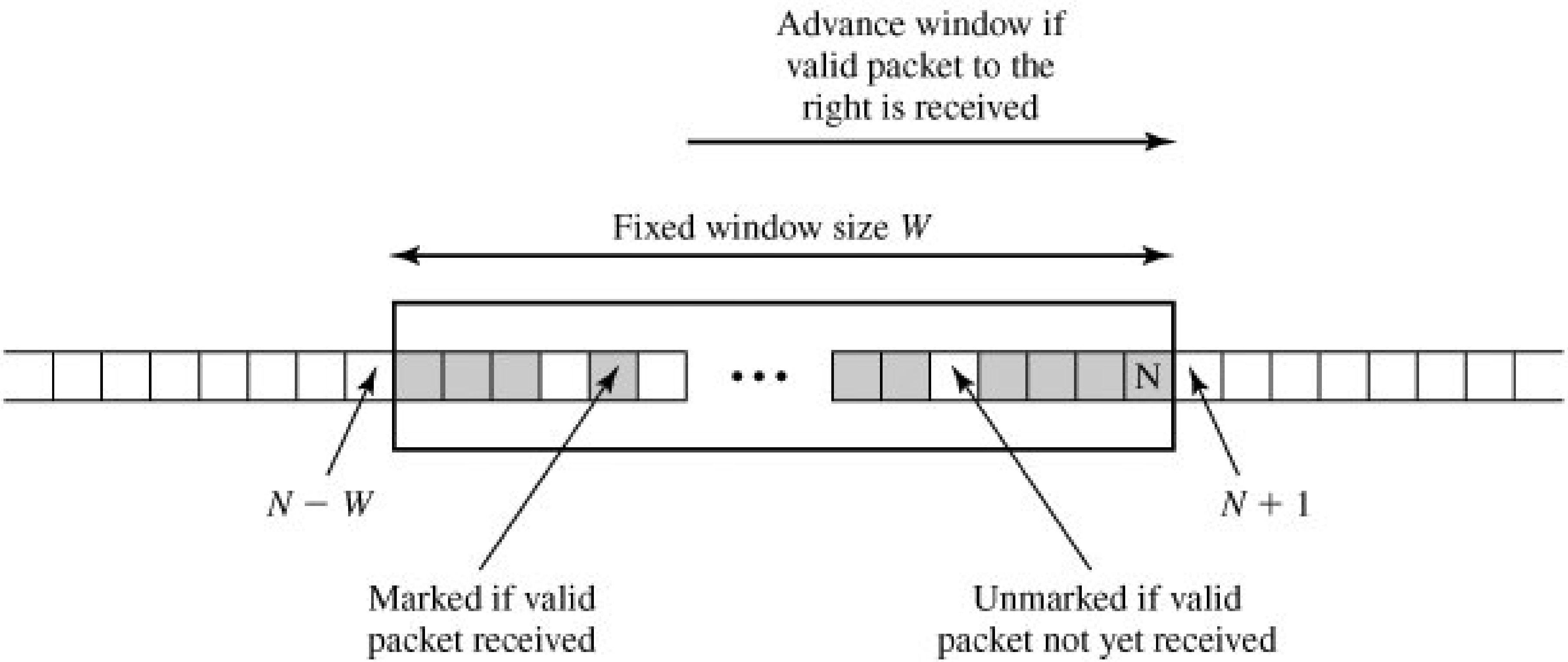
What is hashed?

- Everything that is not mutable during transportation including source and destination addresses.
- Mutable parts are set to all zero before hashing (e.g. time to live, header checksum)
- Authentication is based on the fact that there is a shared key between the two systems.
- The HMAC is called ICV (Integrity Check Value)

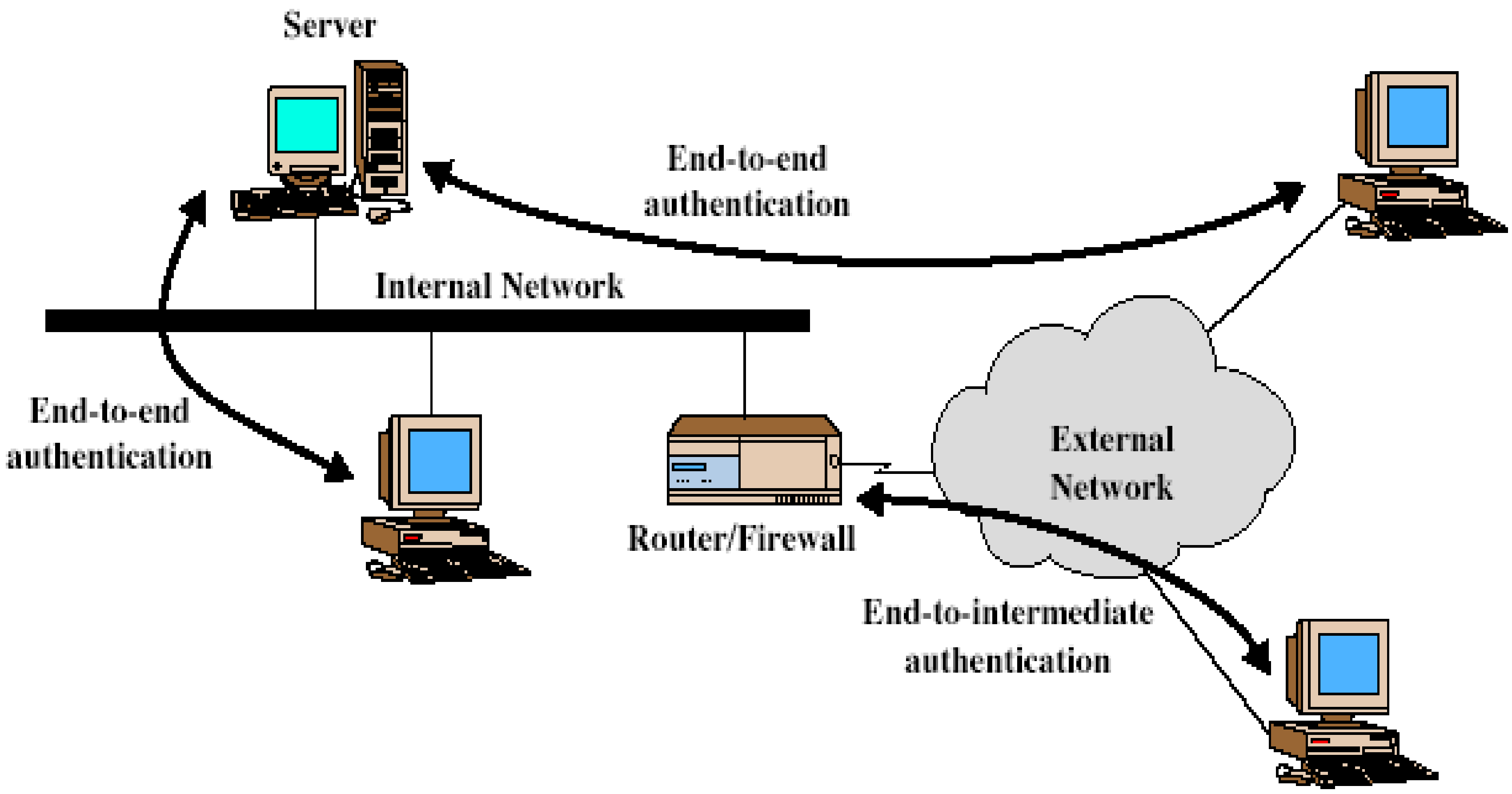
Scope of Authentication



Protection against replay



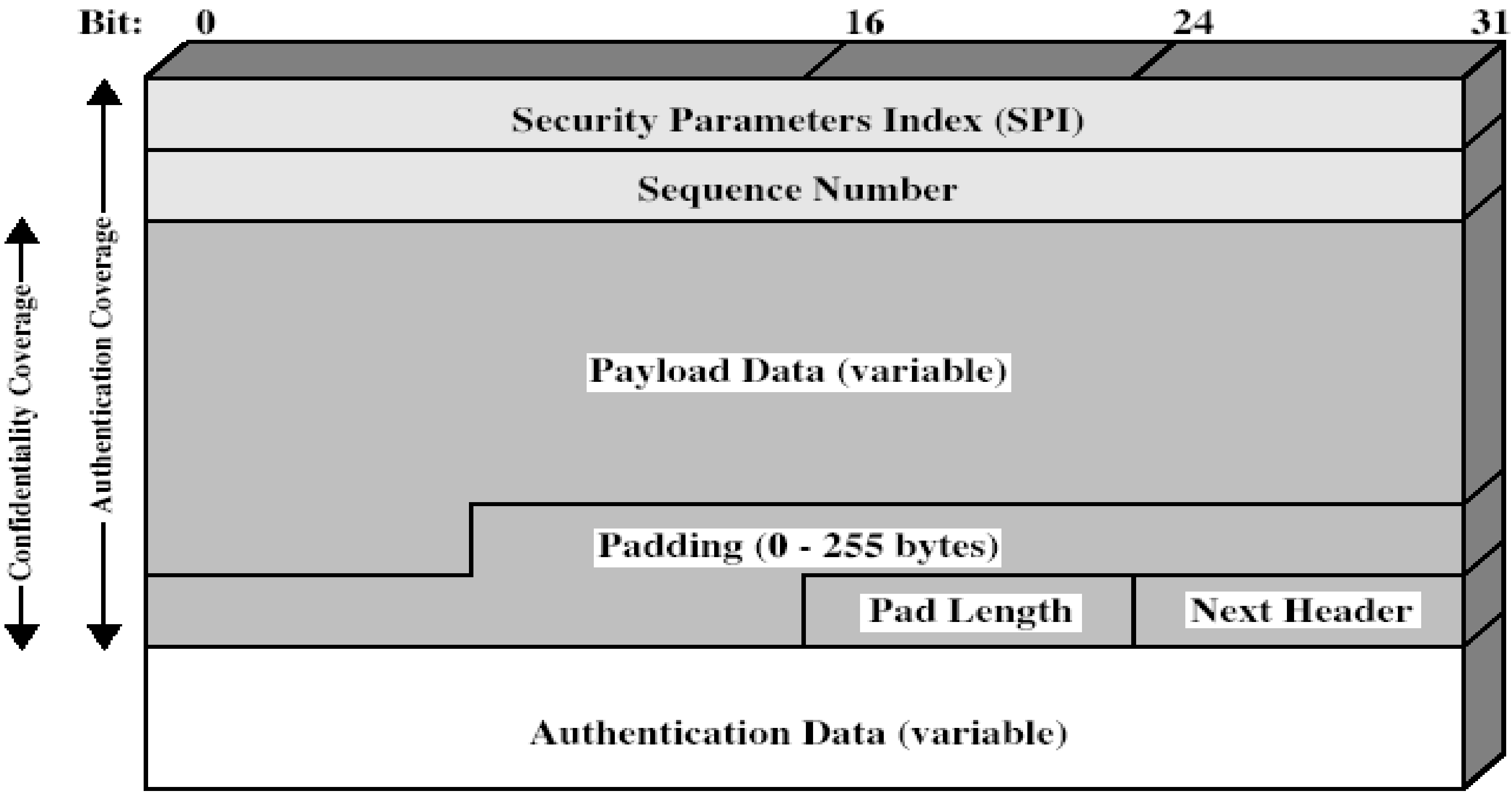
Transport & Tunnel Modes



Encapsulating Security Payload (ESP)

- provides message content confidentiality & limited traffic flow confidentiality
- can optionally provide the same authentication services as AH
- supports range of ciphers, modes, padding
  - incl. DES, Triple-DES, RC5, IDEA, CAST etc
  - CBC most common
  - pad to meet blocksize, for traffic flow

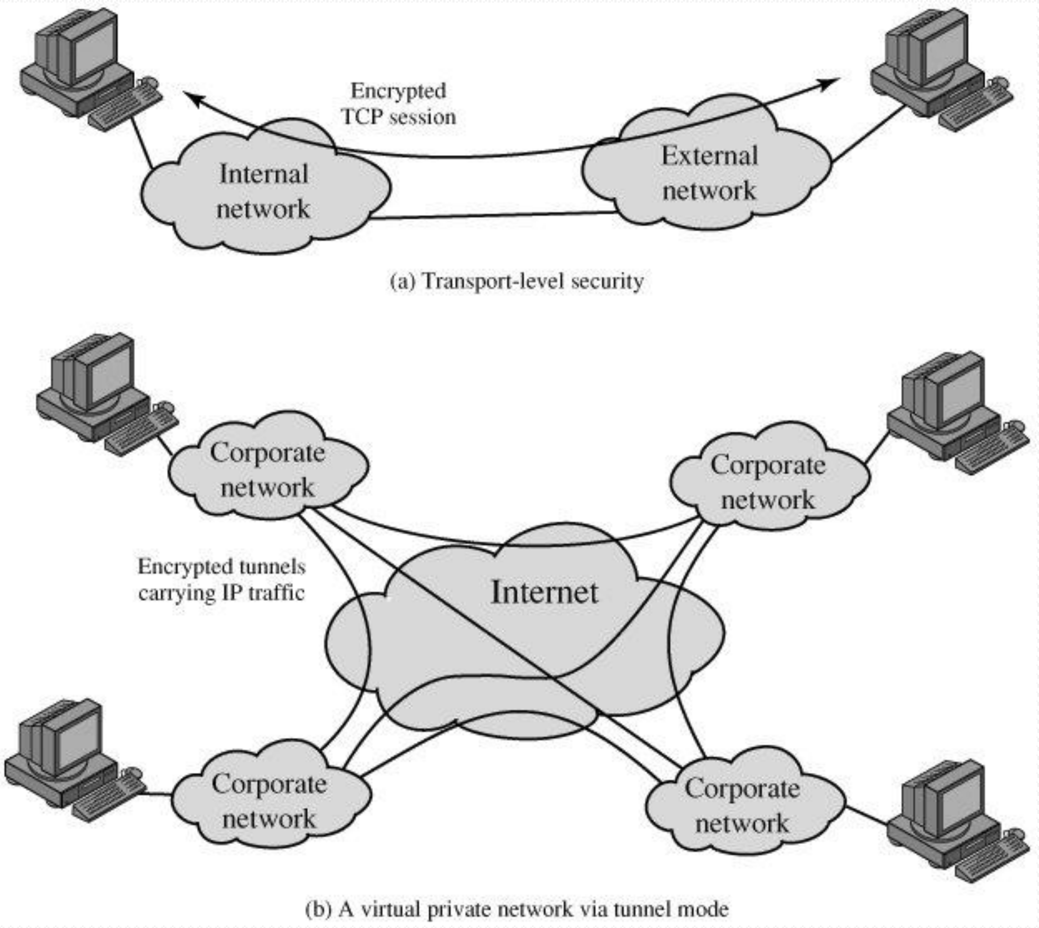
Encapsulating Security Payload



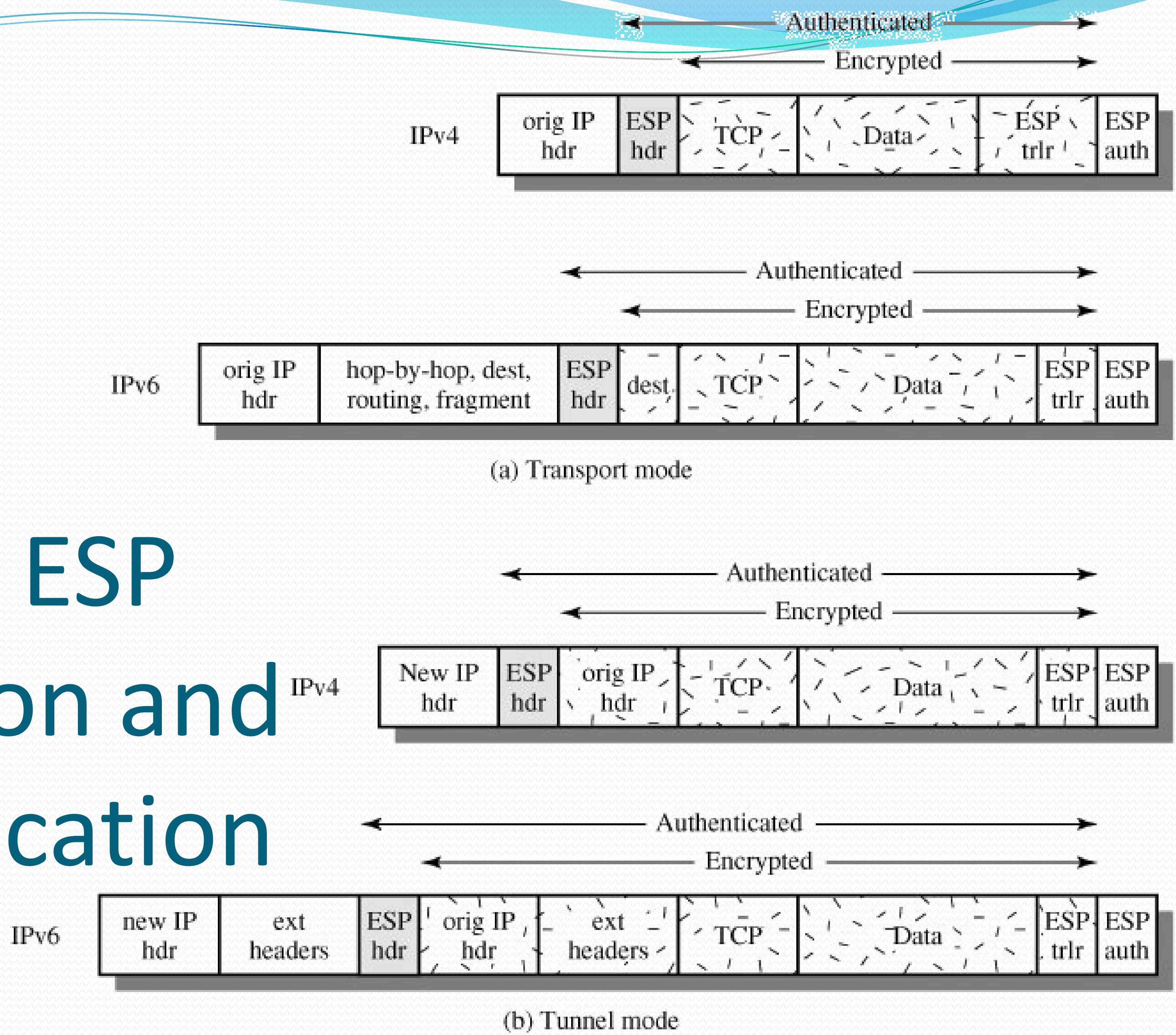
Transport vs Tunnel Mode ESP

- transport mode is used to encrypt & optionally authenticate IP data
  - data protected but header left in clear
  - can do traffic analysis but is efficient
  - good for ESP host to host traffic
- tunnel mode encrypts entire IP packet
  - add new header for next hop
  - good for VPNs, gateway to gateway security

Tunneling and Transport Modes



Scope of ESP Encryption and Authentication



How to get Confidentiality + Authentication?

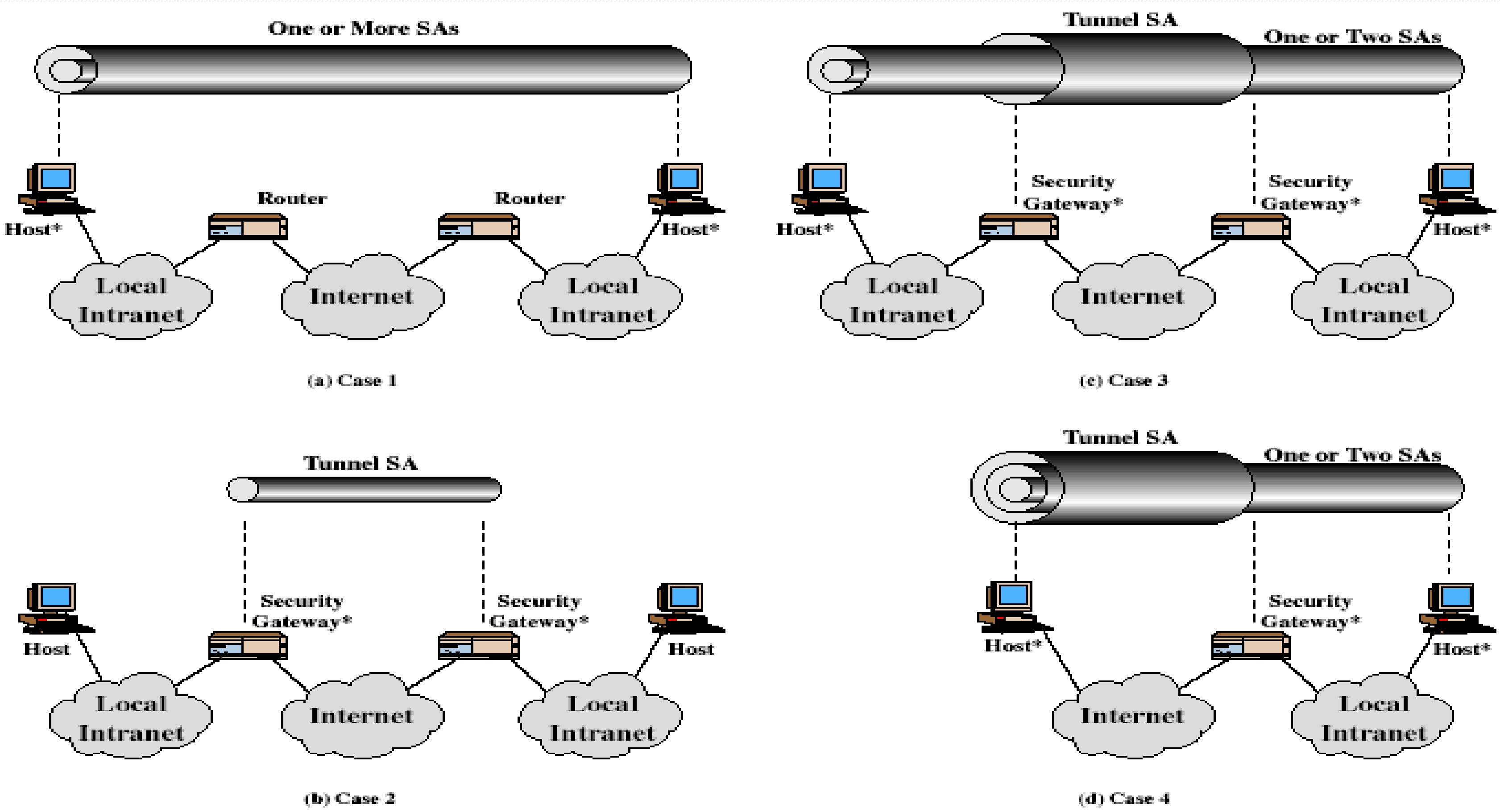
- Single SA: ESP with Authentication Option
  - Transport or Tunnel mode
  - Cannot authenticate some fields in the header including source and destination

- Two SAs: ESP and AH
  - Transport Adjacency (ESP then AH)
    - Both in transport
  - Transport-Tunnel (AH then ESP)
    - AH is in transport and ESP a tunnel
    - Protects authentication data by ESP encryption

Combining Security Associations

- SA's can implement either AH or ESP
- to implement both need to combine SA's
  - form a security bundle
- have 4 cases (see next)

Combining Security Associations



Key Management

- handles key generation & distribution
- typically need 2 pairs of keys
    - 2 per direction for AH & ESP
- manual key management
    - sysadmin manually configures every system
- automated key management
    - automated system for on demand creation of keys for SA's in large systems
    - has Oakley & ISAKMP elements

Oakley

- a key exchange protocol
- based on Diffie-Hellman key exchange
- adds features to address weaknesses
  - cookies, groups (global params), nonces, DH key exchange with authentication
- can use arithmetic in prime fields or elliptic curve fields

Example Oakley Exchange

$\mathbf{I \rightarrow R:}$ CKY_I, OK_KEYX, GRP, g^x, EHAO, NIDP, ID_I, ID_R, N_I, $S_{KI}[ID_I \| ID_R \| N_I \| GRP \| g^x \| EHAO]$

$\mathbf{R \rightarrow I:}$ CKY_R, CKY_I, OK_KEYX, GRP, g^y, EHAS, NIDP, ID_R, ID_I, N_R, N_I, $S_{KR}[ID_R \| ID_I \| N_R \| N_I \| GRP \| g^y \| g^x \| EHAS]$

$\mathbf{I \rightarrow R:}$ CKY_I, CKY_R, OK_KEYX, GRP, g^x, EHAS, NIDP, ID_I, ID_R, N_I, N_R, $S_{KI}[ID_I \| ID_R \| N_I \| N_R \| GRP \| g^x \| g^y \| EHAS]$

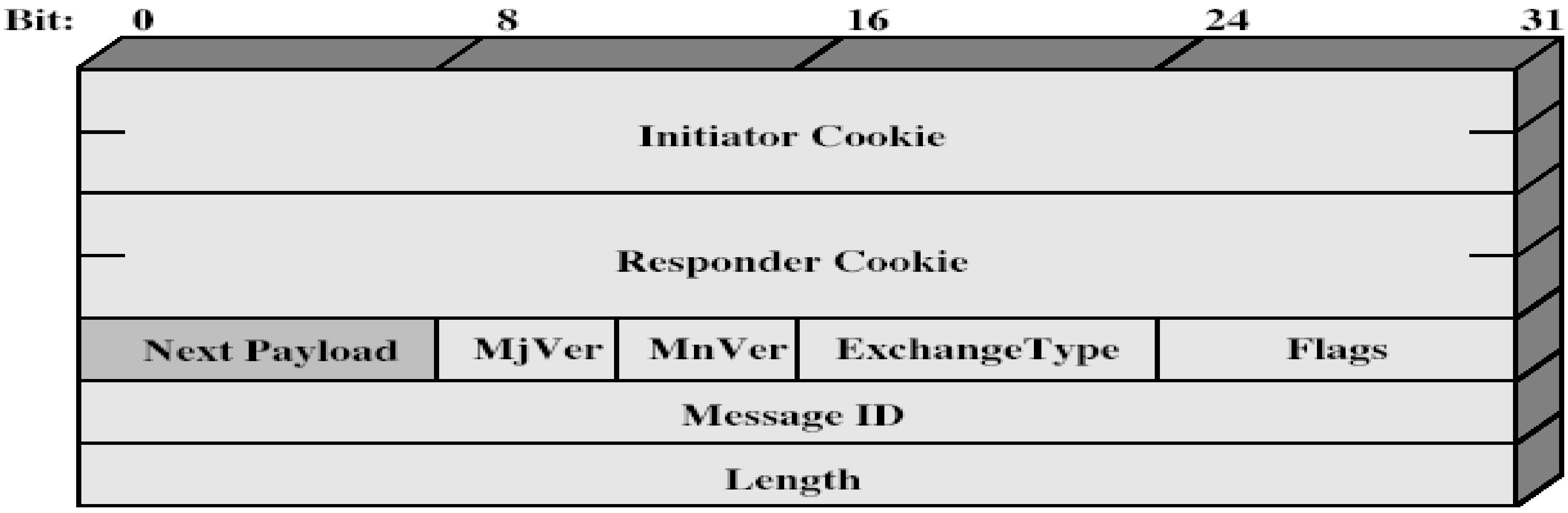
Notation:

I	=	Initiator
R	=	Responder
CKY_I, CKY_R	=	Initiator, responder cookies
OK_KEYX	=	Key exchange message type
GRP	=	Name of Diffie-Hellman group for this exchange
g^x,g^y	=	Public key of initiator, responder; g^{xy} = session key from this exchange
EHAO, EHAS	=	Encryption, hash authentication functions, offered and selected
NIDP	=	Indicates encryption is not used for remainder of this message
ID_I, ID_R	=	Identifier for initiator, responder
N_I, N_R	=	Random nonce supplied by initiator, responder for this exchange
$S_{KI}[X]$, $S_{KR}[X]$	=	Indicates the signature over X using the private key (signing key) of intiator, responder

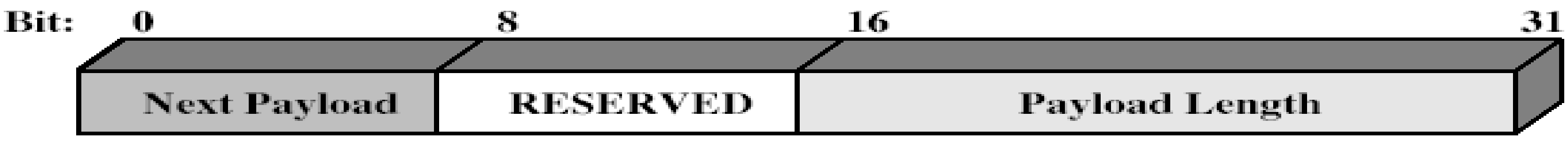
ISAKMP

- Internet Security Association and Key Management Protocol
- provides framework for key management
- defines procedures and packet formats to establish, negotiate, modify, & delete SAs
- independent of key exchange protocol, encryption alg, & authentication method

ISAKMP



(a) ISAKMP Header



(b) Generic Payload Header

ISAKMP Payload Types

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 signature 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.

ISAKMP Exchanges

Exchange	Note
(a) Base Exchange	
(1)**I**⟶**R**: SA; NONCE	Begin ISAKMP-SA negotiation
(2)**R**⟶**E**: SA; NONCE	Basic SA agreed upon
(3)**I**⟶**R**: KE; ID_I AUTH	Key generated; Initiator identity verified by responder
(4)**R**⟶**E**: 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**⟶**E**: SA	Basic SA agreed upon
(3)**I**⟶**R**: KE; NONCE	Key generated
(4)**R**⟶**E**: KE; NONCE	Key generated
(5)*__I__⟶**R**: ID_I; AUTH	Initiator identity verified by responder
(6)*__R__⟶**E**: 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**⟶**E**: 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

I = initiator

R = responder

* = signifies payload encryption after the ISAKMP header

AUTH = authentication mechanism used

ISAKMP Exchanges

(d) Aggressive Exchange

(1)**I**⟶**R**: SA; KE; NONCE; ID_I;	Begin ISAKMP-SA negotiation and key exchange
(2)**R**⟶**E**: 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

I = initiator

R = responder

* = signifies payload encryption after the ISAKMP header

AUTH = authentication mechanism used