CSPC-306

Network Security and Cyber Forensics

IP Security (IPSec protocol)

Dr. Kunwar Pal
NIT Jalandhar

Internetwork Protocol (IP)

Aim

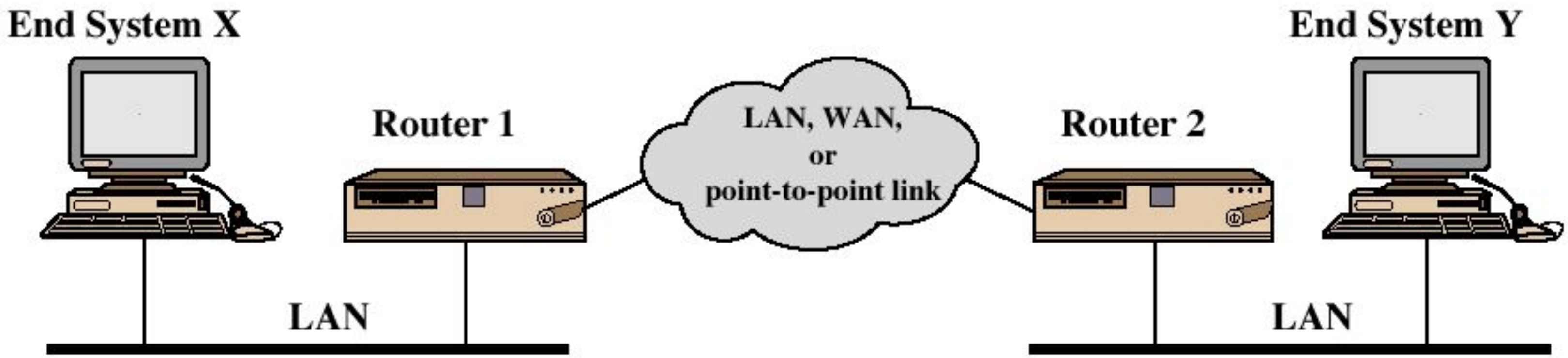
- provide interconnection across different networks

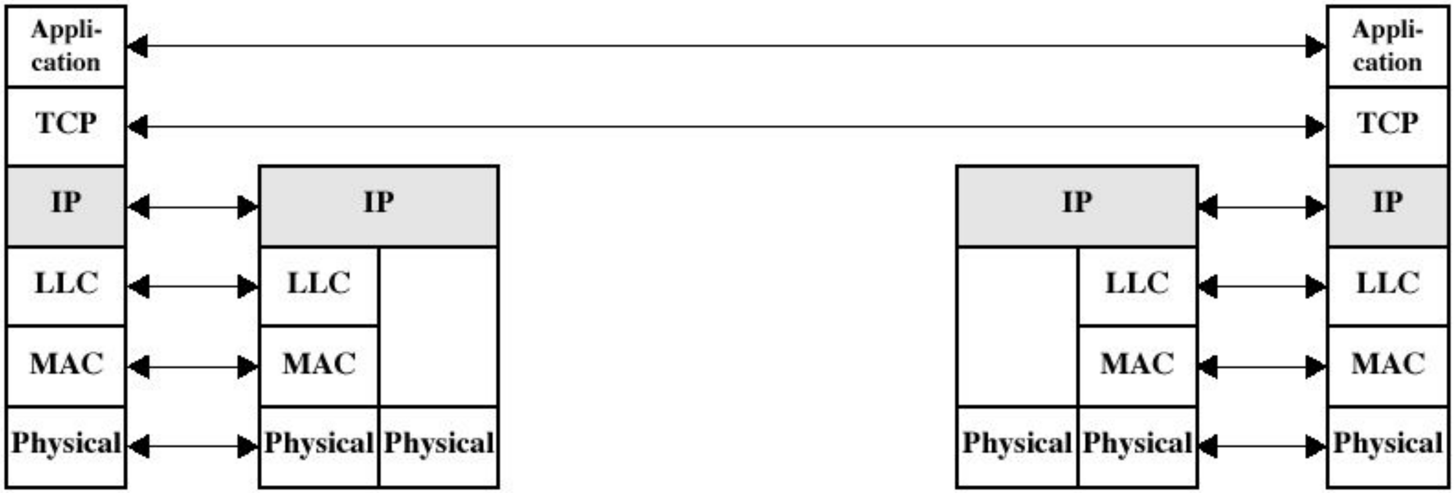
implemented in every end user and in routers

IP is an unreliable protocol

- IP datagrams may be lost
- IP datagrams may arrive out of order
- TCP takes care of those problems

Internetwork Protocol (IP)

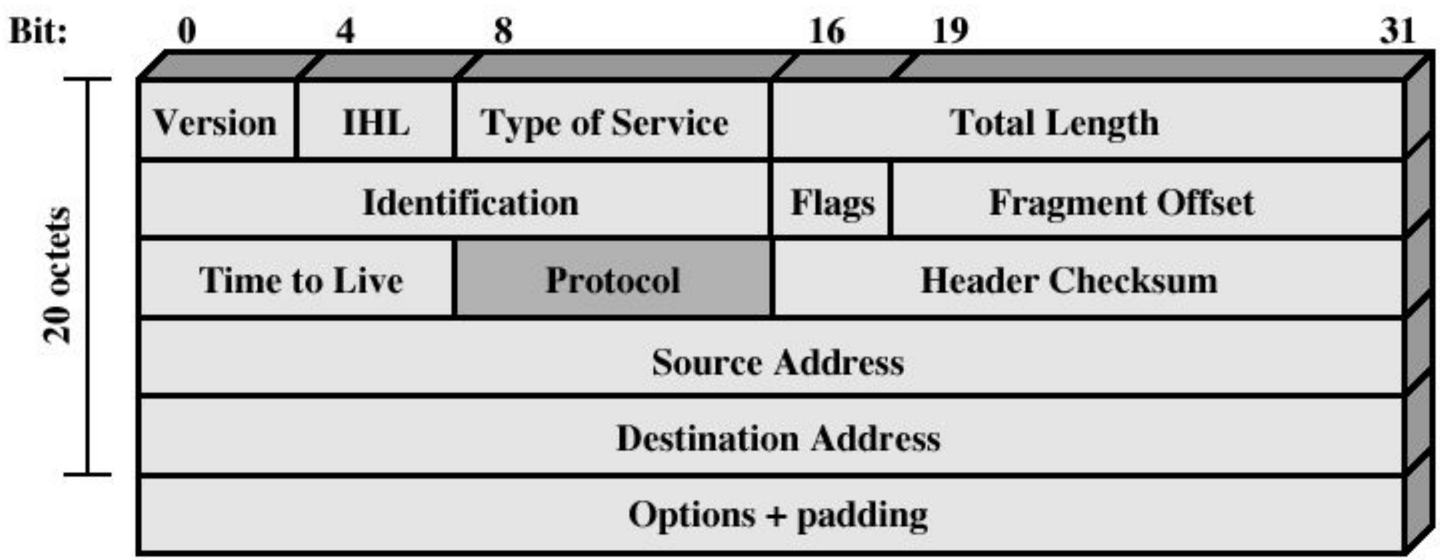




IPv4

The IP version that we are currently using on SU campus

- actually most IP networks are IPv4



(a) IPv4 Header

Data (Payload) follows the header

IPv6

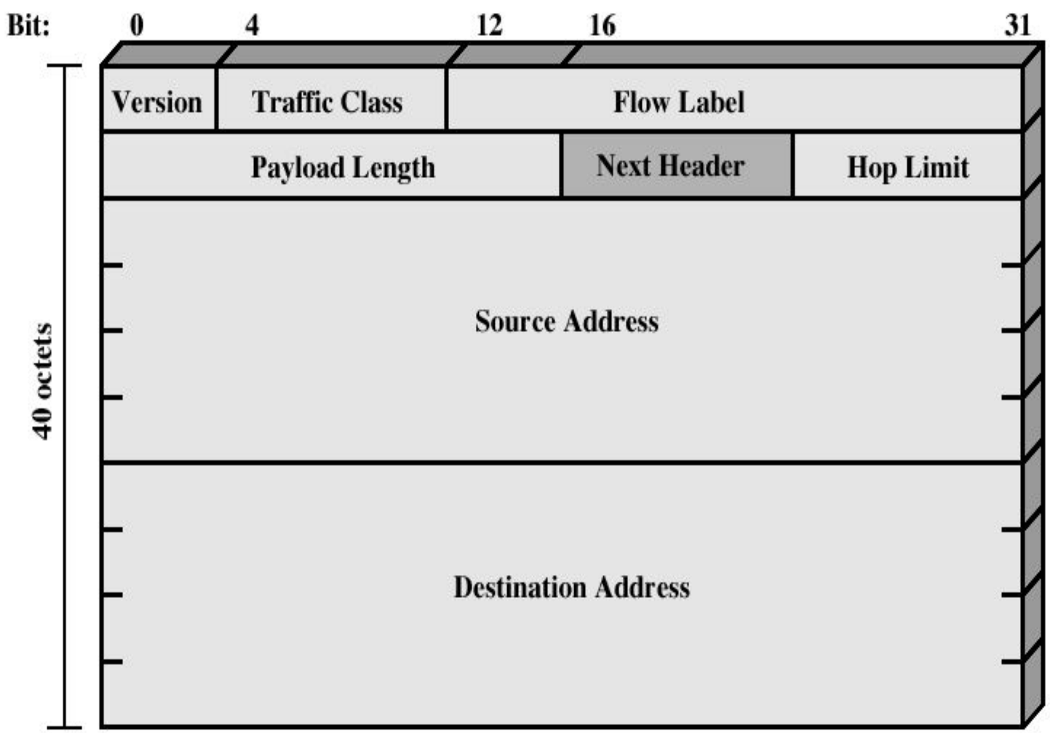
Next generation IP

- driving force was the inadequateness of IPv4 address space

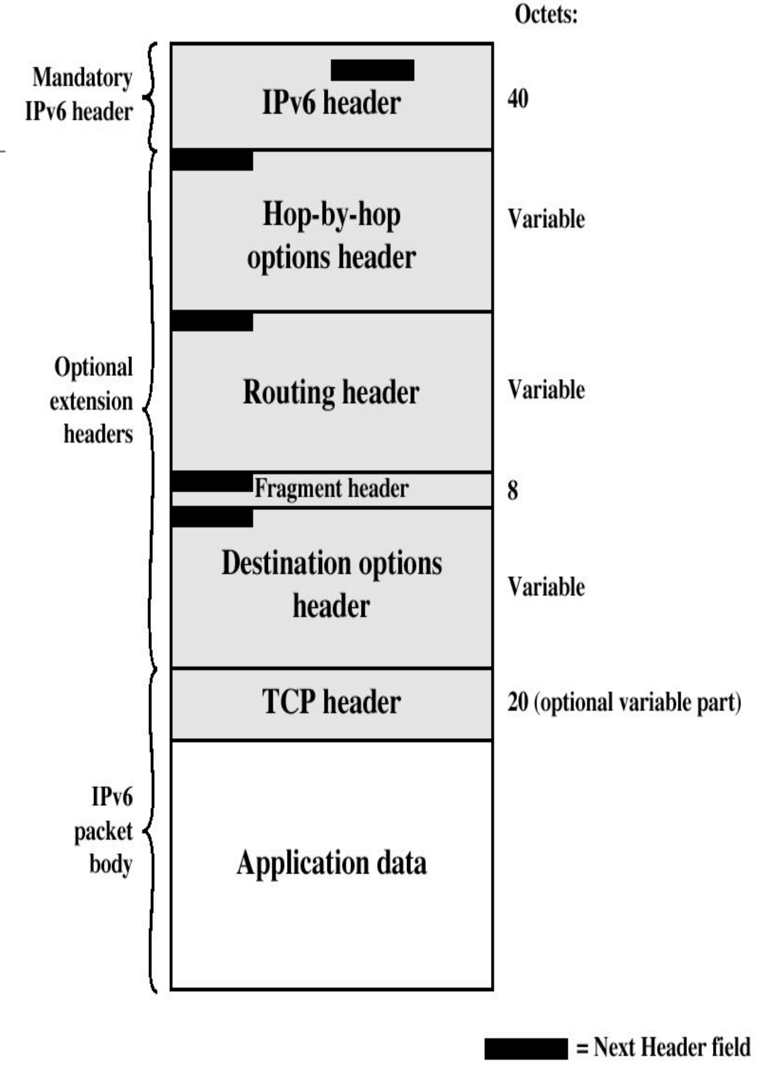
IPv6 header

- modular approach
- base header + extension headers
- base header is longer than v4, but number of fields is smaller

IPv6 header



(b) IPv6 Header



Is IP Secure?

Content (Payload) is not encrypted

- confidentiality is not provided
- IP sniffers are available on the net

IP addresses may be spoofed

- authentication based on IP addresses can be broken

So IP is not secure

Where to provide security?

Application-layer?

- S/MIME, PGP – email security
- Kerberos – client / server
- SSH – secure telnet

Transport level?

- SSL / TLS
- between TCP and Application

IP level

- IPSec

IPSec

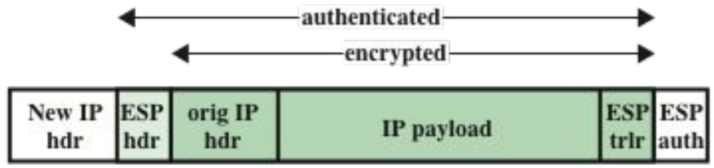
general IP Security mechanisms

provides authentication and confidentiality at IP level
- also has key management features

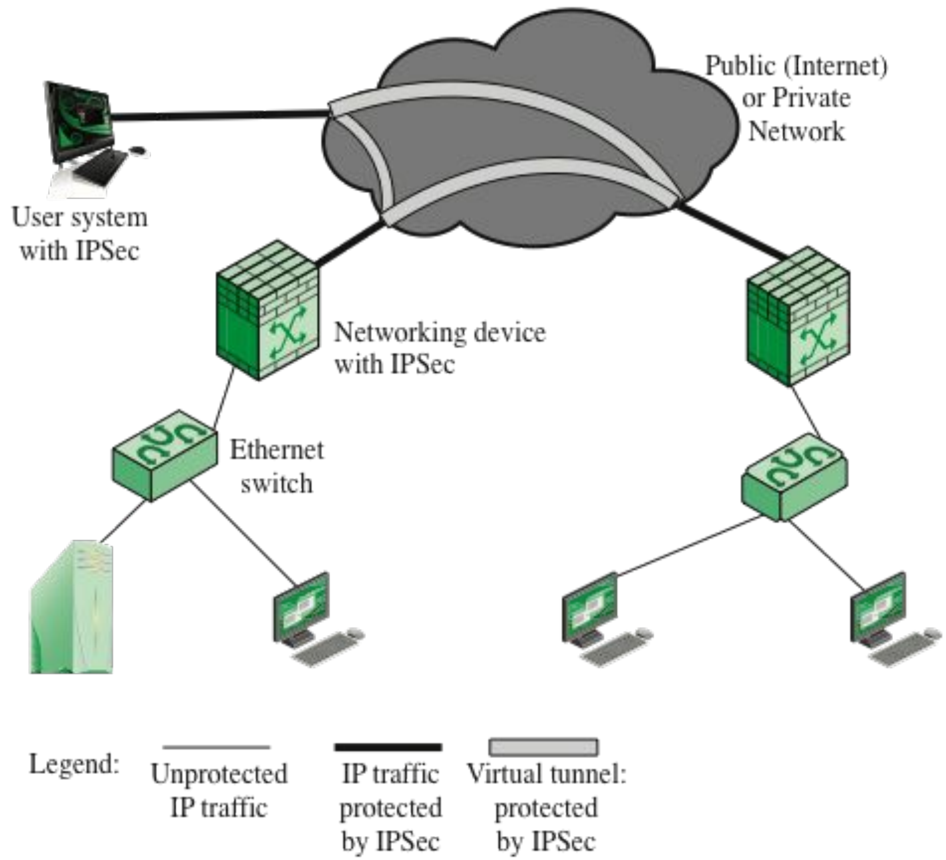
Applications
- VPNs (Virtual Private Networks)
  - Interconnected LANs over the insecure Internet
  - router-to-router
- Secure remote access, e.g. to ISPs
  - individual-to-router

IPSec support is mandatory for IPv6 products, optional for v4
- many manufacturers support IPSec in their v4 products

IPSec Application Scenarios



(a) Tunnel-mode format



(b) Example configuration

Benefits of IPSec

in a firewall/router, IPSec provides strong security to all traffic entering the network

- without passing the security overhead to the internal network and workstations
- user transparent: no need to assume security-aware users, no per-user keys

IPSec is below transport layer

- transparent to applications
- No need to upgrade applications when IPSec is used, if IPSec is implemented and configured in user machines

IPSec Documentation and Standards

IPSec and its specifications are quite complex

defined in numerous RFCs (6071)

- most important RFCs are 4301 (Overview of security architecture), 4302 (AH - Authentication Header), 4303 (ESP – Encapsulating Security Payload – for encryption), 7296 (IKEv2 – Key Management)
- many others, see IETF IPSec Working Group website
  - http://datatracker.ietf.org/wg/ipsec/charter/

IPSec Protocols

Authentication Header (AH)

- defines the authentication protocol
- no encryption
- Since ESP covers authentication, it is not recommended anymore
  - But we will talk about it

Encapsulating Security Payload (ESP)

- provides encryption
- optionally authentication

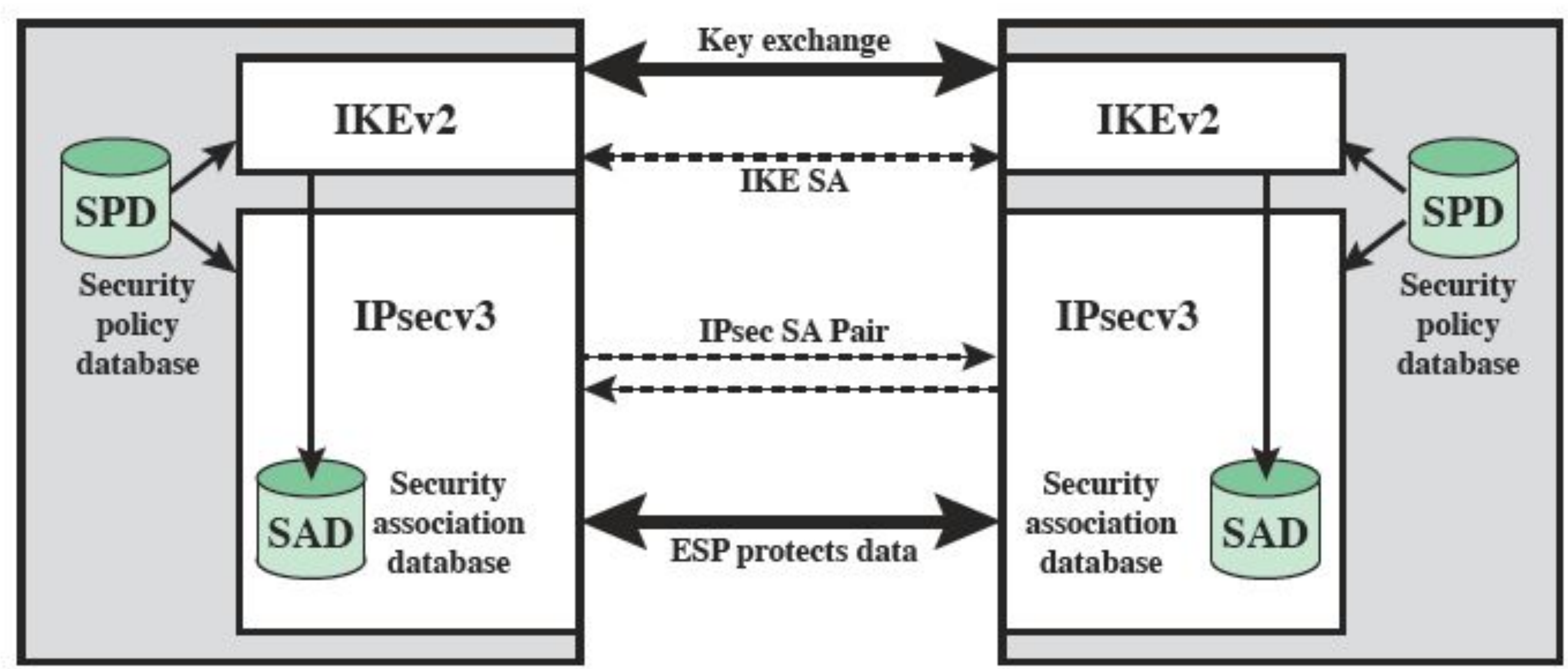
Crypto algorithms that support those protocols are generally defined in other documentation

Key distribution and management are also in different RFCs

IPSec Services

	AH	ESP (encryption only)	ESP (encryption plus authentication)
Access control	✔	✔	✔
Connectionless integrity	✔		✔
Data origin authentication	✔		✔
Rejection of replayed packets	✔	✔	✔
Confidentiality		✔	✔
Limited traffic flow confidentiality		✔	✔

IPSec General Architecture (Big Picture)



Security Associations (SA)

a one-way relationship between sender & receiver

- specifies IPSec related parameters

Identified by 3 parameters:

- Destination IP Address
- Security Protocol: AH or ESP
- Security Parameters Index (SPI)
    - A local 32-bit identifier (to be carried later to endpoints within AH and ESP)

There are several other parameters associated with an SA

- stored locally in Security Association Databases (SAD)

SA Parameters (some of them)

Anti-replay related

- Sequence Number Counter
  - to generate sequence numbers
- Anti-replay window
  - something like sliding-window; will be discussed later.

AH info

- authentication algorithms, keys, key lifetimes, etc.

ESP info

- encryption (and authentication) algorithms, keys, key lifetimes, etc.

Lifetime of SA

IPSec Mode: Transport or Tunnel

SA, AH – ESP, and key management

SAs are in databases

- both in sender and receiver

AH and ESP use the cryptographic primitives and other info in SA

Key Management Protocols (will discuss later) are to establish SA

So

- AH / ESP are independent of key management

SA Selectors

IPSec is a flexible protocol

- ~~traffic from IP address X to IP address Y may use several SAs~~
  - or no SA if that particular traffic will not be secured

Security Policy Database (SPD) is used to assign a particular IP traffic to an SA

- fields of an SPD entry are called selectors

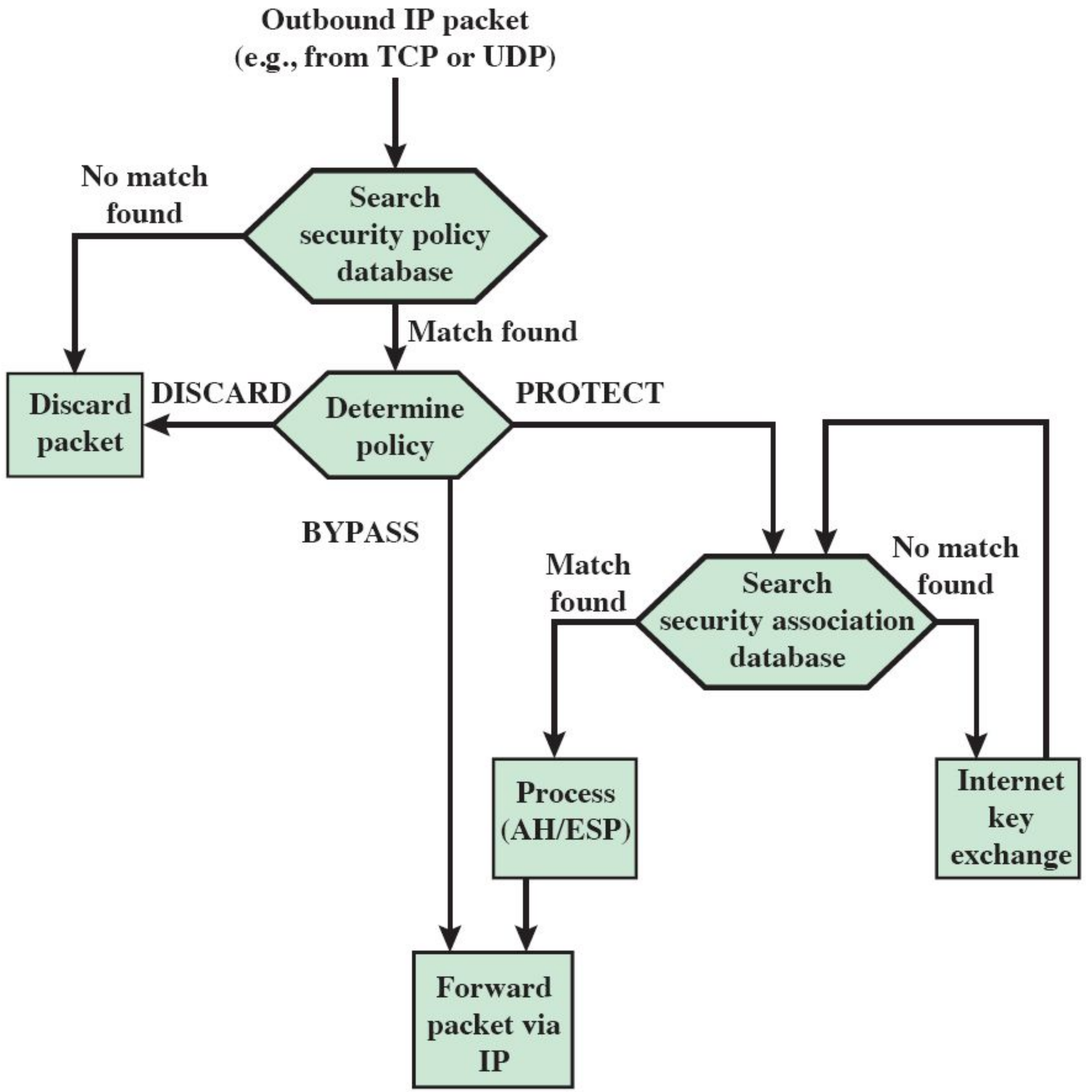
Outbound processing

- compare the selector fields of SPD with the one in the IP traffic
- Determine the SA, if any
- If there exists an SA, do the AH or ESP processing

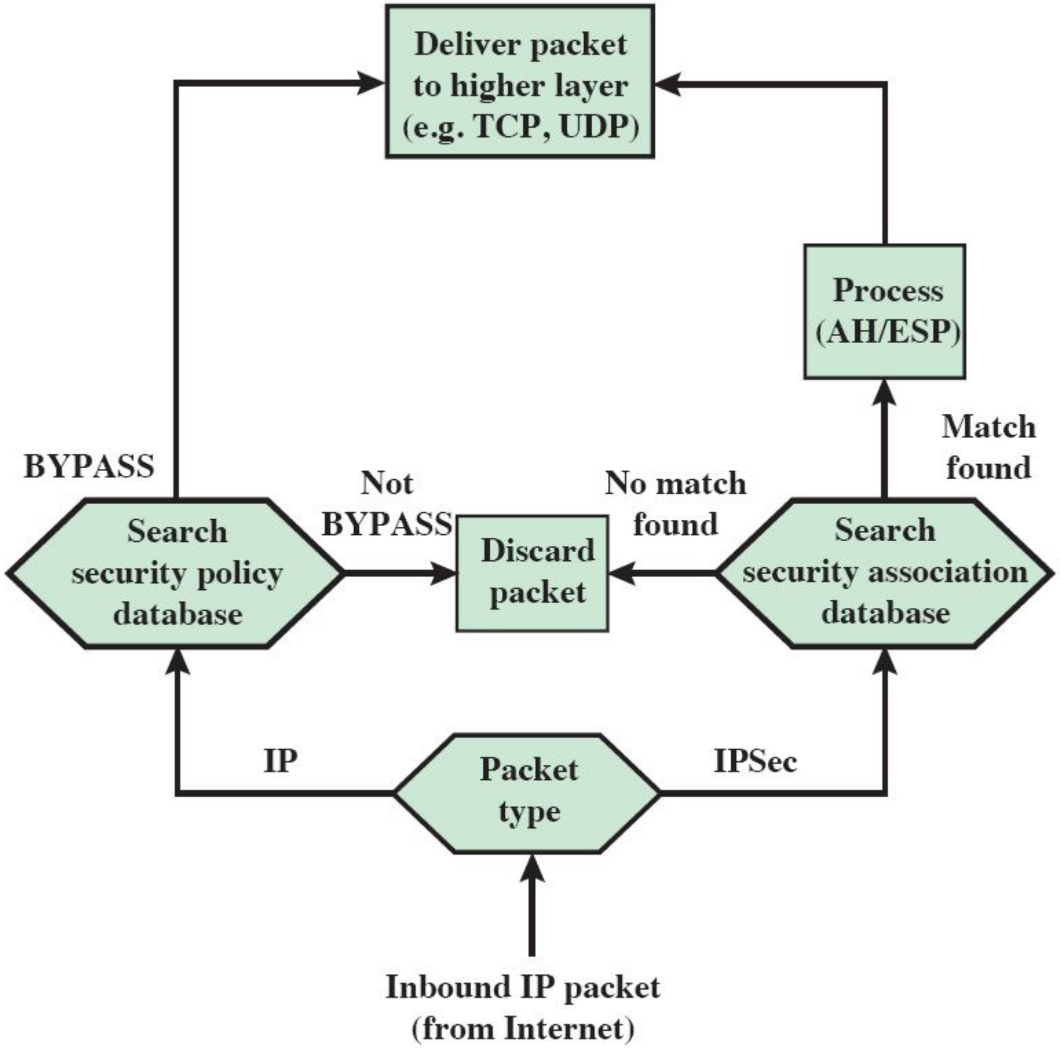
Inbound processing

- Check the incoming IPSec packet and process with AH or ESP
- Discard in case of an anomaly

Outbound Processing Model



Inbound Processing Model



Some SA Selectors

Destination and Source IP addresses

- range, list and wildcards allowed

Transport Layer Protocol

- TCP, UDP, ICMP, all

Source and Destination Ports

- list and wildcards allowed
- from TCP or UDP header

etc.

Host (IP Addr: 1.2.3.101) SPD Example

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

Transport and Tunnel Modes

Both AH and ESP support these two modes

- differently (will see later)

Transport Mode

- security is basically for the IP payload (upper-level protocol data)
- IP header is not protected (except some fields in AH)
- Typically for end-to-end communication

Tunnel Mode

- secures the IP packet as a whole incl. header(s)
- actually puts all IP packet within another (outer) one
- packet is delivered according to the outer IP header
- Typically for router-to-router, or firewall-to-firewall communication

Authentication Header (AH)

Provides support for data integrity and authentication of IP packets

- malicious modifications are detected
- address spoofing is prevented
- replays are detected via sequence numbers

Authentication is based on use of a MAC

- parties must share a secret key
  - in SA

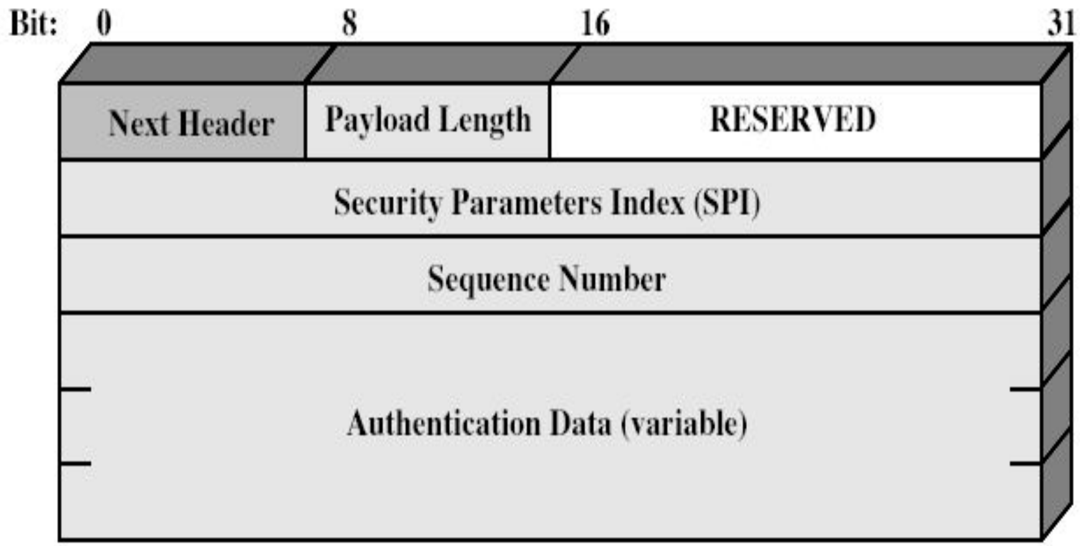
Authentication Header

Next Header: specifies next header or upper layer protocol

Payload length: to specify header length

SPI: to identify SA

Sequence number: used for replay control



Authentication data: MAC value (variable length)

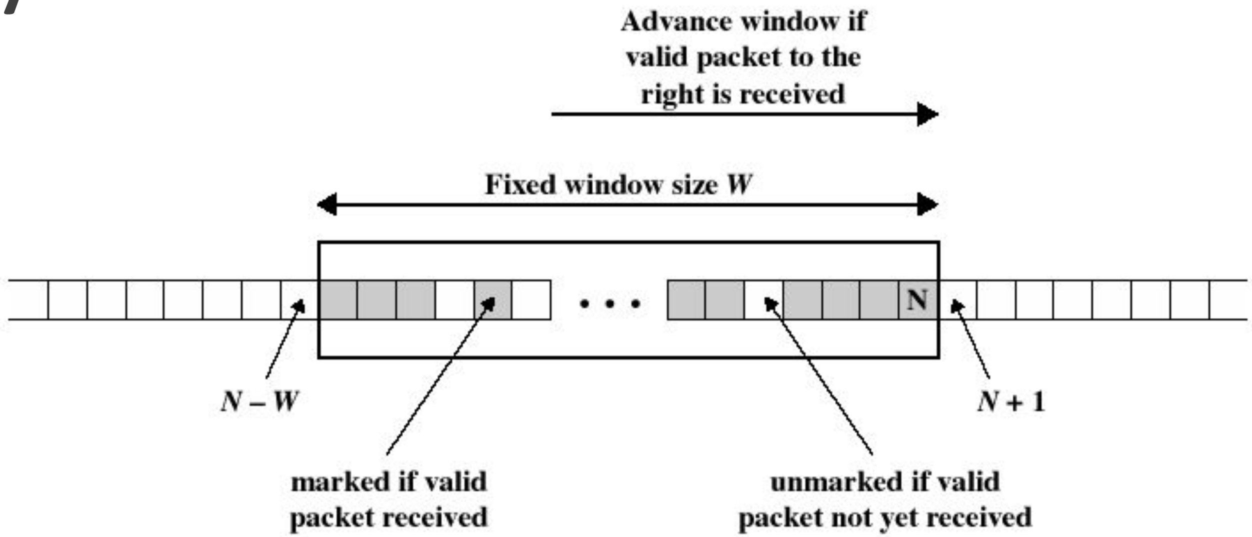
AH – Anti-replay Service

Detection of duplicate packets

Sequence numbers
- associated with SAs
- 32-bit value
- when an SA is created, initialized to 0
  - when it reaches 2^{32}-1, SA must be terminated
  - not to allow overflows
- sender increments the replay counter and puts into each AH (sequence number field)

Problem: IP is unreliable, so the receiver may receive IP packets out of order
- Solution is window-based mechanism
  - Implemented at receiver side

Anti-replay Service



window size W
(default is 64)

N: highest seq. number for a valid paket recevied so far

If a received packet falls in the window
- if authenticated and unmarked, mark it
- if marked, then replay!

If a received packet is > N
- if authenticated, advance the window so that this packet is at the rightmost edge and mark it

If a received packet is <= N-W
- packet is discarded

AH - Integrity Check Value (ICV)

Actually it is a MAC

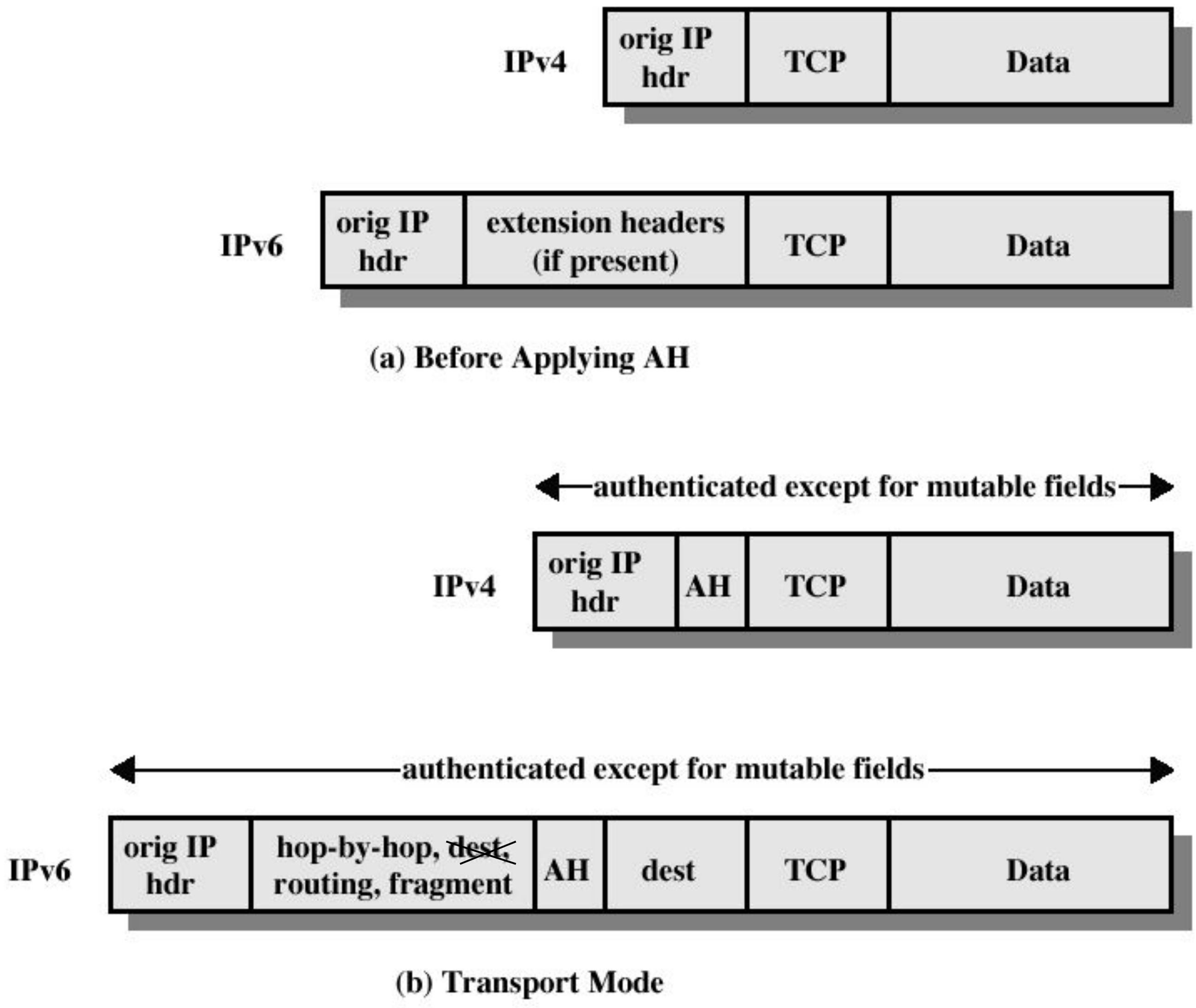
HMAC is used

- with a secure hash algorithm
- default length of authentication data field is 96
  - so HMAC output is truncated

MAC is calculated over

- IP payload (upper layer protocol data)
- IP Headers that are "immutable" or "mutable but predictable" at destination
  - e.g. source address (immutable), destination address (mutable but predictable)
  - Time to live field is mutable. Such mutable fields are zeroed for MAC calculation
- AH header (except authentication data of course, since authentication data is the MAC itself)

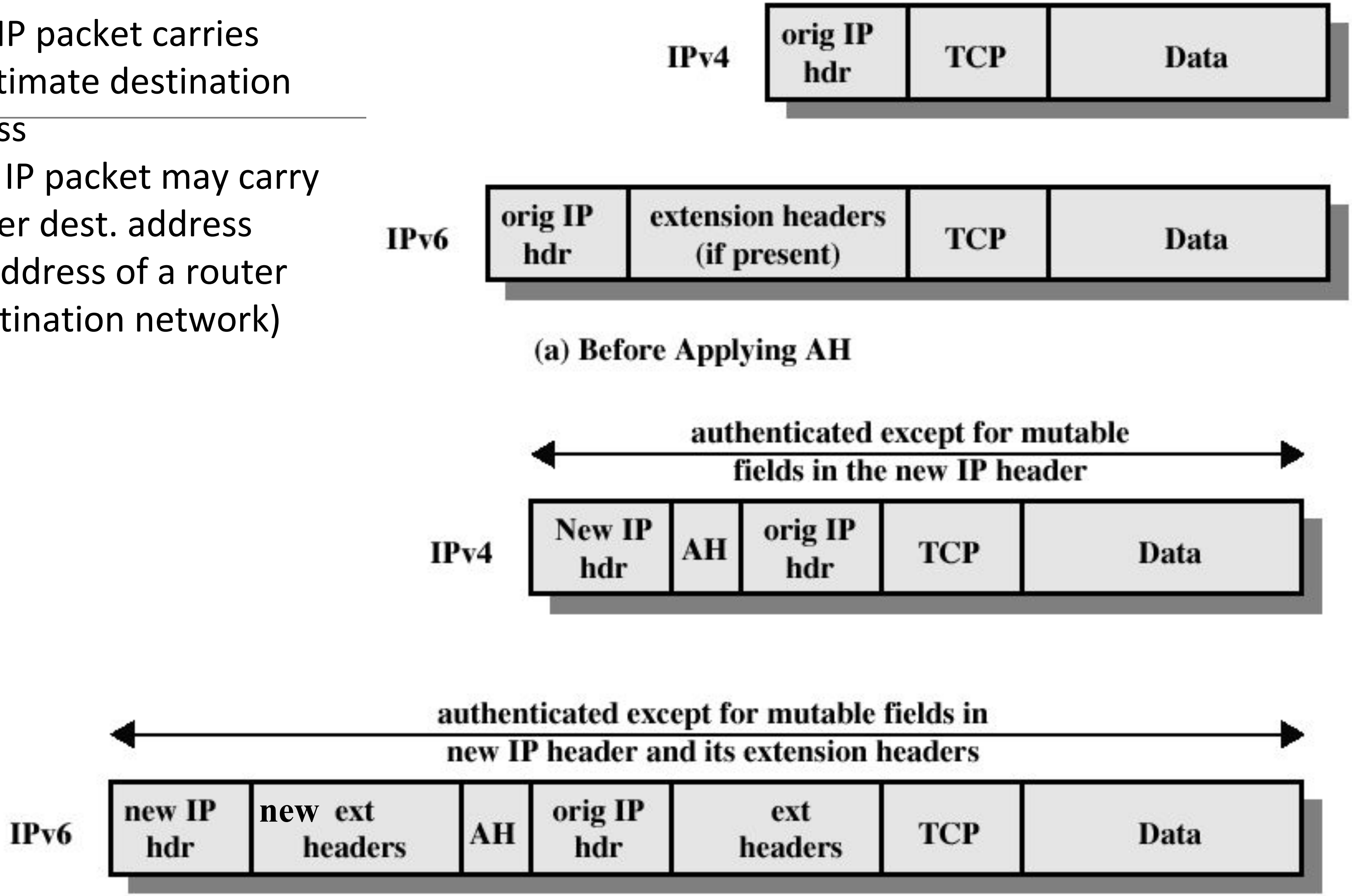
AH – Transport Mode



AH – Tunnel Mode

Inner IP packet carries the ultimate destination address

Outer IP packet may carry another dest. address (e.g. address of a router at destination network)



(c) Tunnel Mode

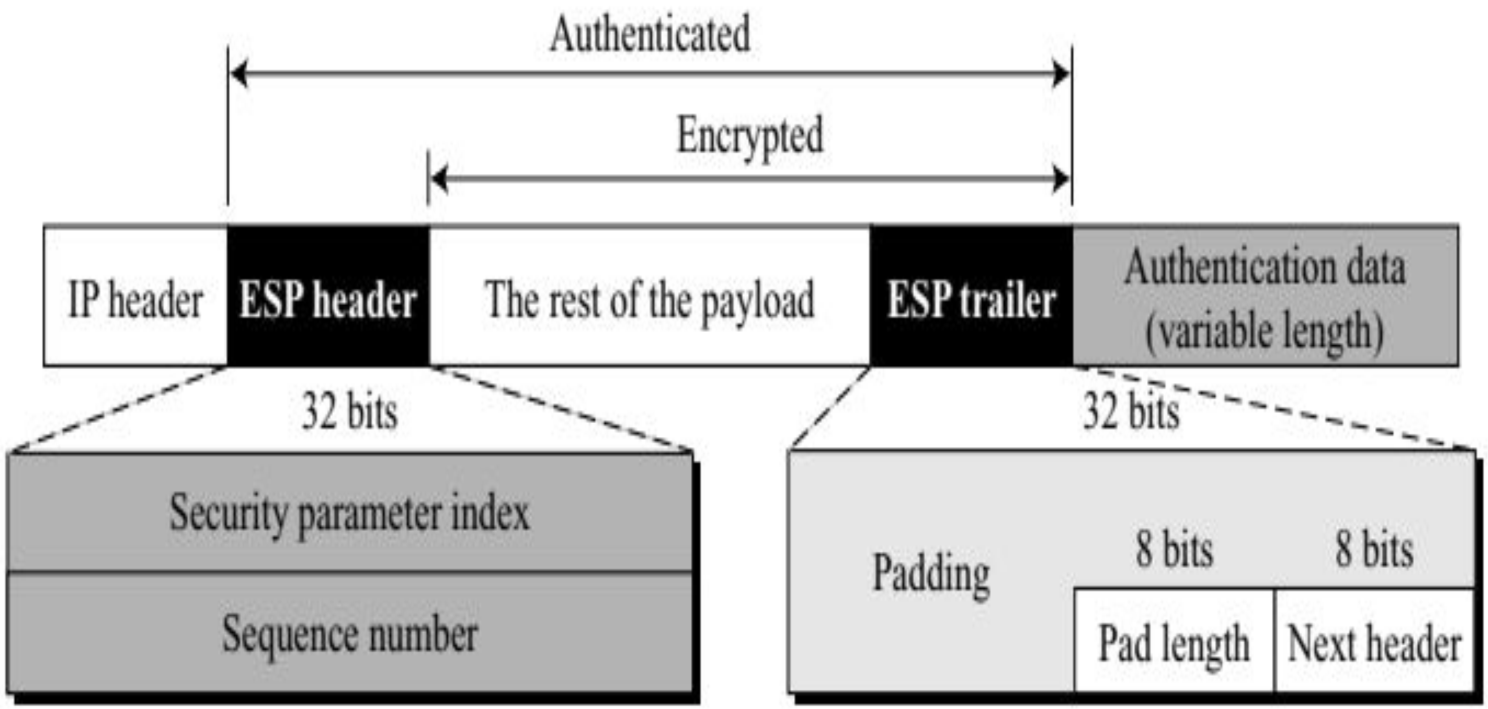
Encapsulating Security Payload (ESP)

provides

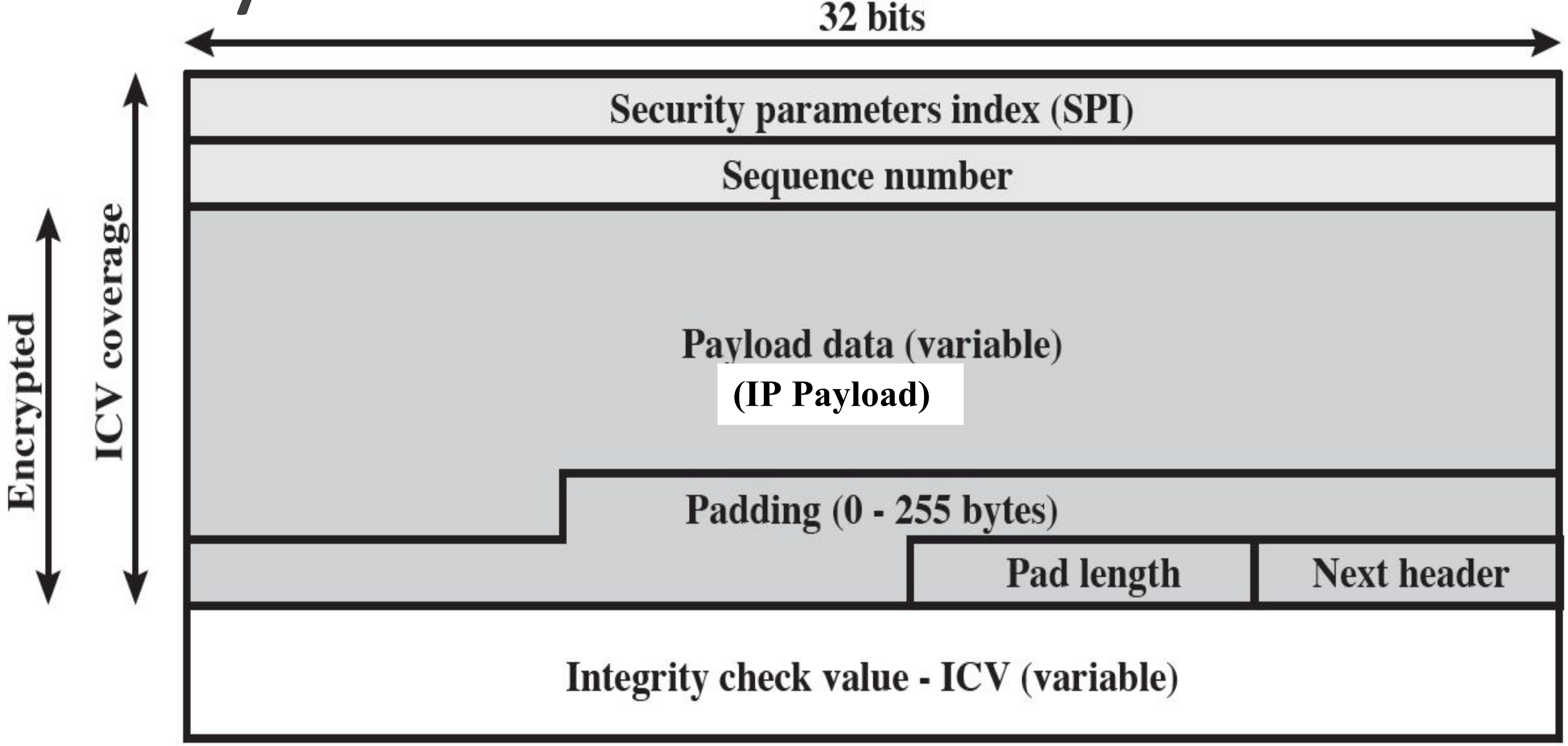
- message content confidentiality
  - via encryption
- limited traffic flow confidentiality and measures for traffic analysis
  - by padding (may arbitrarily increase the data)
  - by encrypting the source and destination addresses in tunnel mode
- optionally authentication services as in AH
  - via MAC (HMAC), sequence numbers

supports range of ciphers, modes

- DES, Triple-DES, RC5, IDEA, Blowfish, etc.
- CBC is the most common mode


Authenticated
Encrypted
IP header
ESP header
The rest of the payload
ESP trailer
Authentication data
(variable length)
32 bits
32 bits
Security parameter index
Sequence number
Padding
8 bits
8 bits
Pad length
Next header

Encapsulating Security Payload



Padding in ESP

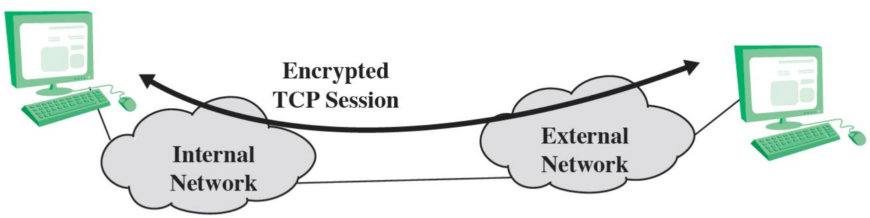
several purposes and reasons

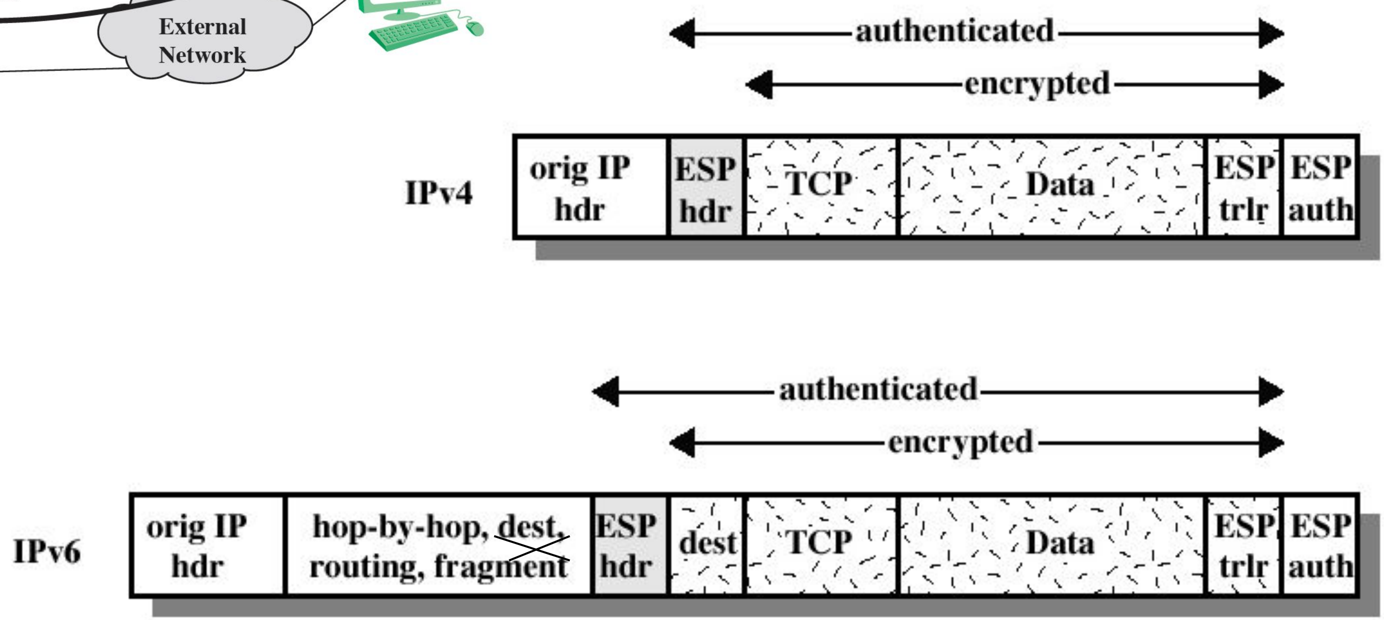
- encryption algorithm may require the plaintext to be multiple of some integer *n*
- ESP format requires 32-bit words
- additional padding may help to provide partial traffic flow confidentiality by concealing the actual length of data
  - Other than the existing padding field, extra padding can be added to the end of the payload to improve traffic flow confidentiality

Transport Mode ESP

transport mode is used to encrypt & optionally authenticate IP payload (e.g. TCP segment)

- data protected but IP header left in clear
- so source and destination addresses are not encrypted
- Mostly for host to host (end-to-end) traffic



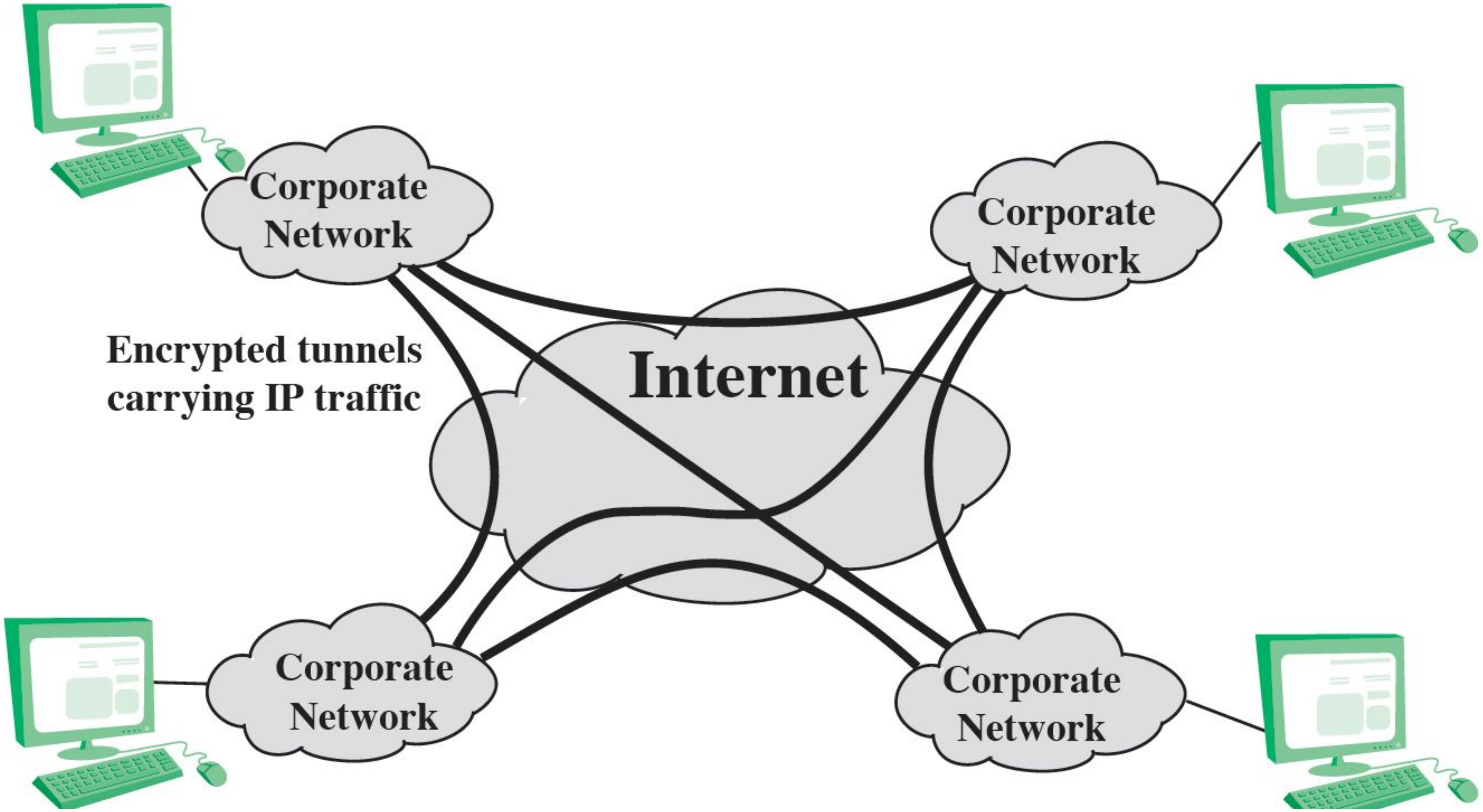


Tunnel Mode ESP

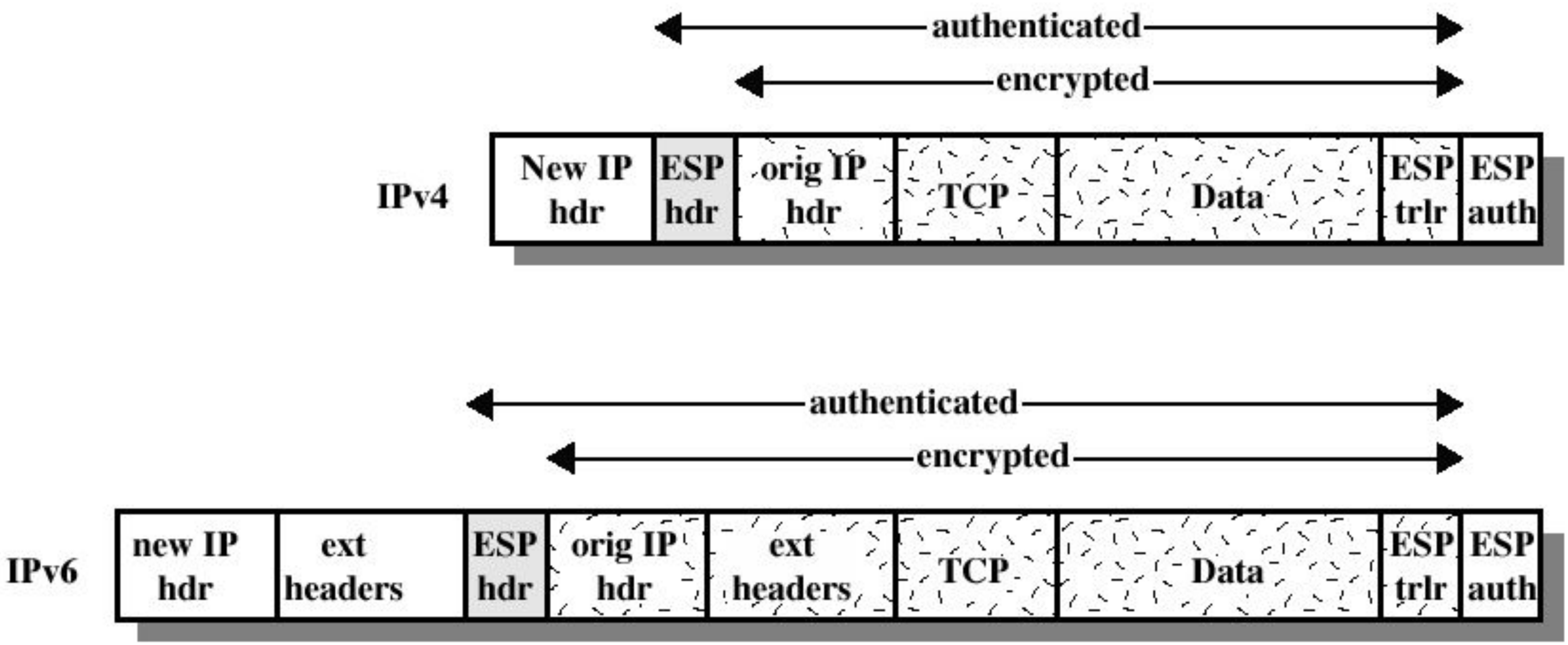
Encrypts and optionally authenticates the entire IP packet

- add new (outer) IP header for processing at intermediate routers
  - may not be the same as the inner (original) IP header, so traffic analysis can somehow be prevented
- good for VPNs, gateway to gateway (router to router) security
  - hosts in internal network do not get bothered with security related processing
  - number of keys reduced
  - thwarts traffic analysis based on ultimate destination

Tunnel Mode ESP

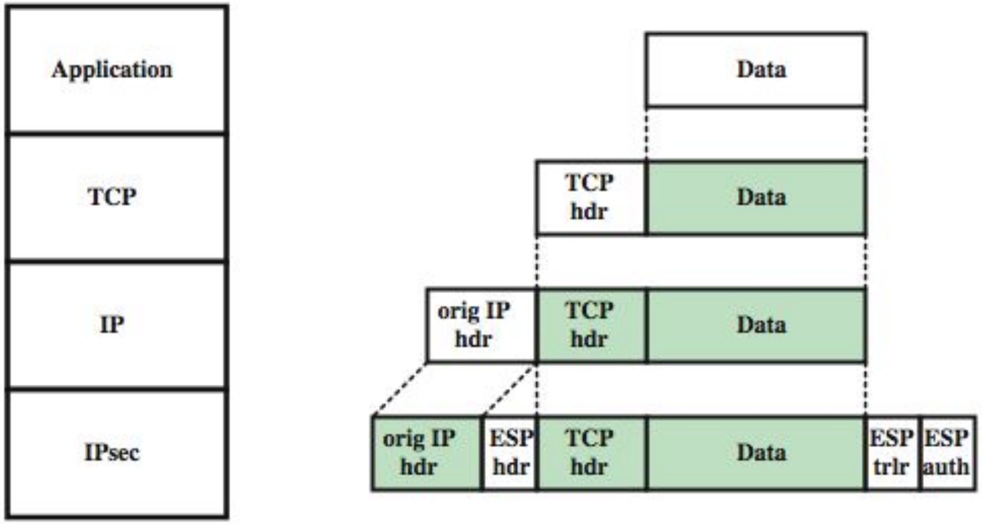


Tunnel Mode ESP

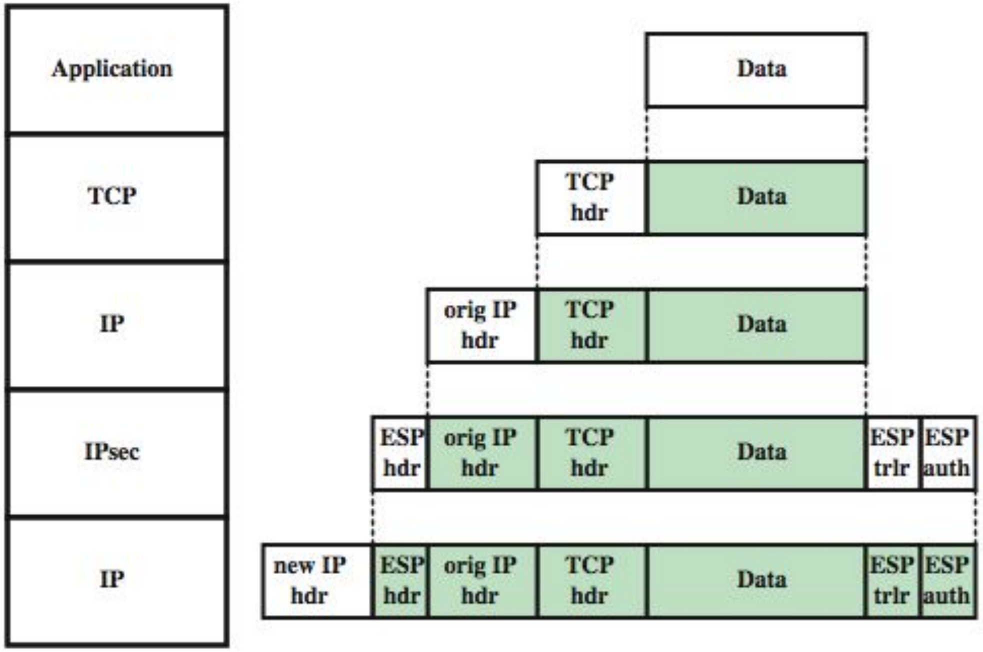


(b) Tunnel Mode

Protocol Operations for ESP



(a) Transport mode



(b) Tunnel mode

Transport and Tunnel Modes

	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.

Combining Security Associations

SAs can implement either AH or ESP

to implement both, need to combine SAs

- form a security association bundle

A possible case: End-to-end Authentication + Confidentiality

- Solution1: use ESP with authentication option on
- Solution2: apply ESP SA (no auth.) first, then apply AH SA
- Solution3: Apply AH SA first, then ESP SA
  - encryption is after the authentication

Internet Key Exchange

The key management portion of IPsec involves the determination and distribution of secret keys

- A typical requirement is four keys for communication between two applications
  - Transmit and receive pairs for both integrity and confidentiality

The IPsec Architecture document mandates support for two types of key management:

Manual

- A system administrator manually configures each system with its own keys and with the keys of other communicating systems
- This is practical for small, relatively static environments

Automated

- Enables the on-demand creation of keys for SAs and facilitates the use of keys in a large distributed system with an evolving configuration

ISAKMP/Oakley

The default automated key management protocol of IPsec

Consists of:

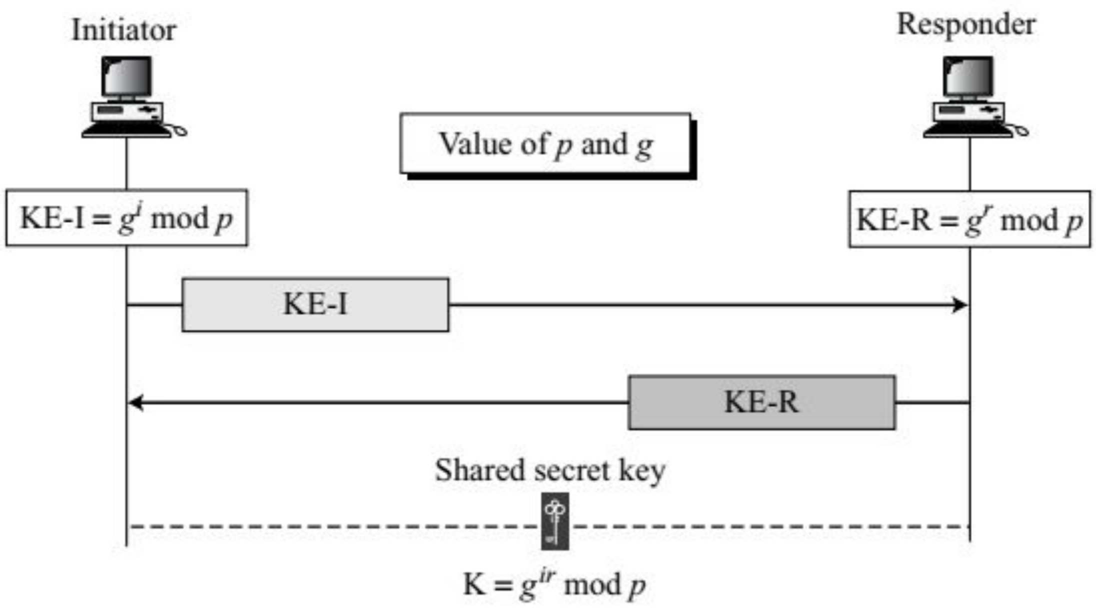
- Oakley Key Determination Protocol
  - A key exchange protocol based on the Diffie-Hellman algorithm but providing added security
  - Generic in that it does not dictate specific formats
- Internet Security Association and Key Management Protocol (ISAKMP)
  - Provides a framework for Internet key management and provides the specific protocol support, including formats, for negotiation of security attributes
  - Consists of a set of message types that enable the use of a variety of key exchange algorithms

Features of IKE Key Determination

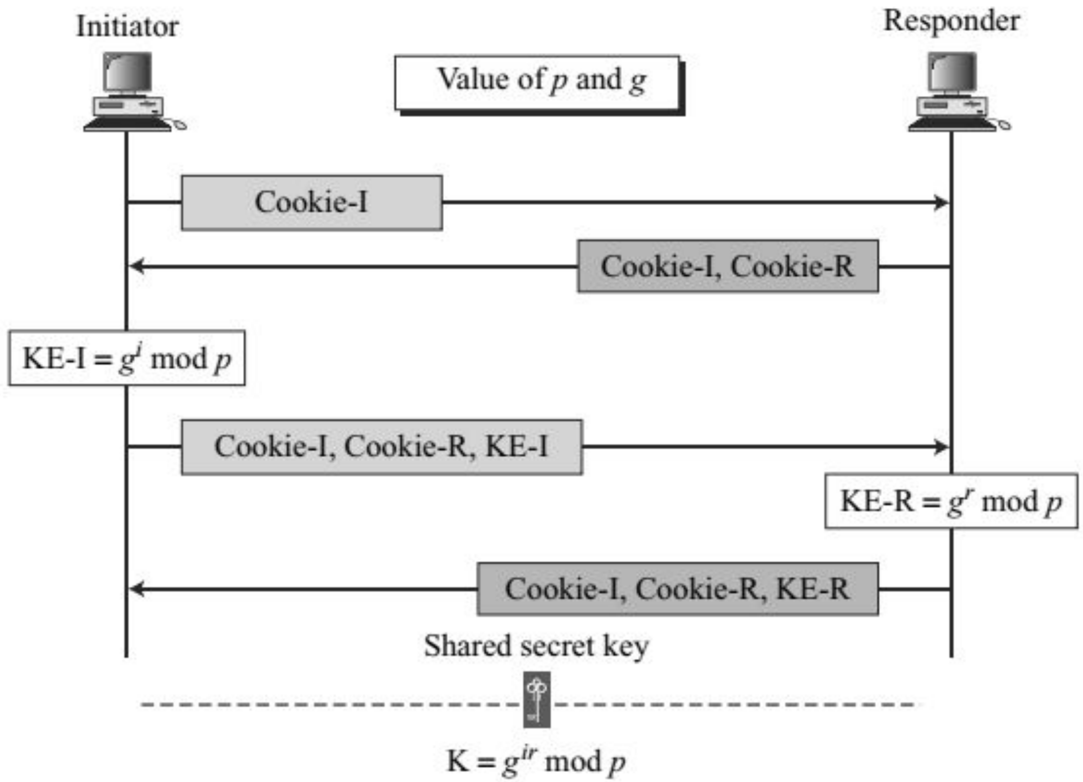
Algorithm is characterized by five important features:

1. It employs a mechanism known as cookies to prevent clogging attacks
2. It enables the two parties to negotiate a group; this, in essence, specifies the global parameters of the Diffie-Hellman key exchange
3. It uses nonces to ensure against replay attacks
4. It enables the exchange of Diffie-Hellman public key values
5. It authenticates the Diffie-Hellman exchange to prevent man-in-the-middle-attacks

Diffie-Hellman key exchange

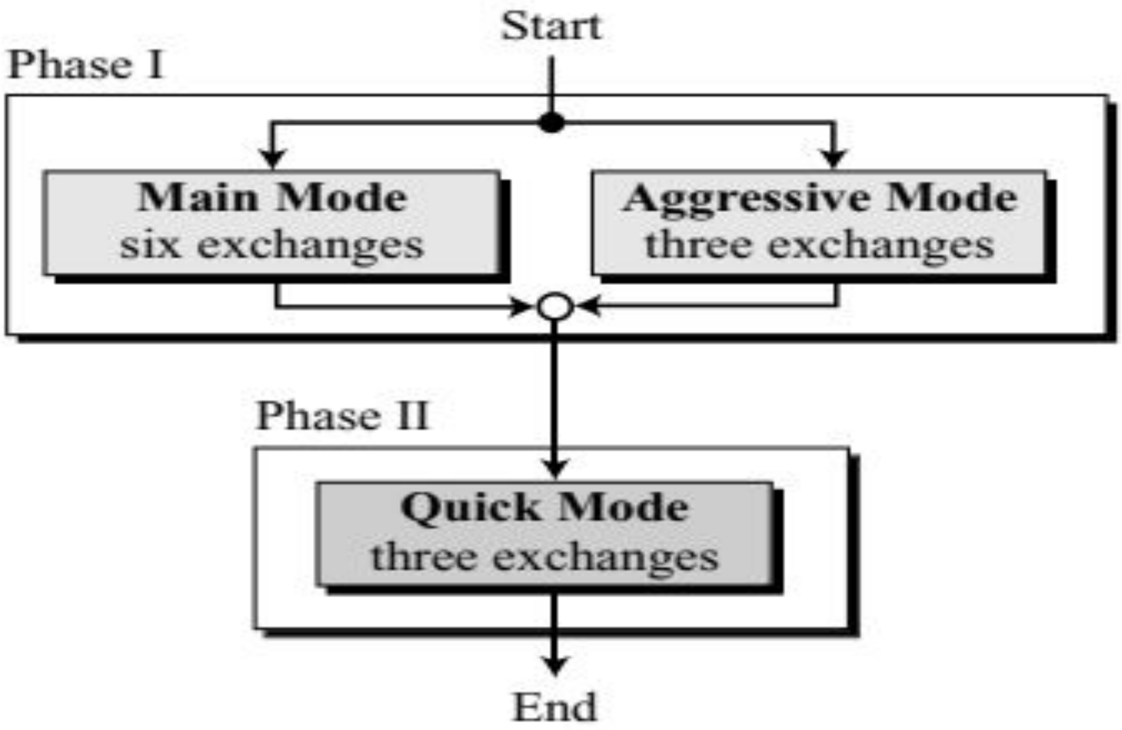


Diffie-Hellman with cookies

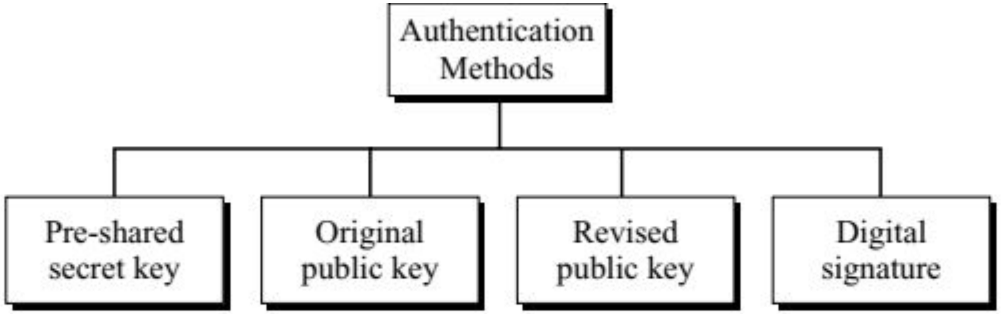


IKE Phases

IKE is divided into two phases: phase I and phase II. Phase I creates SAs for phase II; phase II creates SAs for a data exchange protocol such as IPSec


Start
Phase I
Main Mode
six exchanges
Aggressive Mode
three exchanges
Phase II
Quick Mode
three exchanges
End

Main-mode or aggressive-mode methods



Phase I: Main Mode

- In the main mode, the initiator and the responder exchange six messages.
- first two messages, they exchange cookies (to protect against a clogging attack) and negotiate the SA parameters.
- The initiator sends a series of proposals; the responder selects one of them.
- When the first two messages are exchanged, the initiator and the responder know the SA parameters and are confident that the other party exists (no clogging attack occurs).
- Third and fourth messages, the initiator and responder usually exchange their half-keys (gi and gr of the Diffie-Hellman method) and their nonces (for replay protection). In some methods other information is exchanged.
- Note that the half-keys and nonces are not sent with the first two messages because the two parties must first ensure that a clogging attack is not possible.

- After exchanging the third and fourth messages, each party can calculate the common secret between them in addition to its individual hash digest.
- The common secret SKEYID (secret key ID) is dependent on the calculation method as shown below.
- In the equations, prf (pseudorandom function) is a keyed-hash function defined during the negotiation phase.
- SKEYID_d (derived key) is a key to create other keys.
- SKEYID_a is the authentication key and SKEYID_e is used for the encryption key; both are used during the negotiation phase.
- The first parameter (SKEYID) is calculated for each key-exchange method separately. The second parameter is a concatenation of various data. Note that the key for prf is always SKEYID.
- The two parties also calculate two hash digests, HASH-I and HASH-R, which are used in three of the four methods in the main mode.

$SKEYID = \boldsymbol{prf}\,(\text{preshared-key, N-I} \mid \text{N-R})$ (preshared-key method)

$SKEYID = \boldsymbol{prf}\,(\text{N-I} \mid \text{N-R}, g^{ir})$ (public-key method)

$SKEYID = \boldsymbol{prf}\,(\text{hash (N-I} \mid \text{N-R), Cookie-I} \mid \text{Cookie-R})$ (digital signature)

$SKEYID\_d = \boldsymbol{prf}\,(\text{SKEYID}, g^{ir} \mid \text{Cookie-I} \mid \text{Cookie-R} \mid 0)$

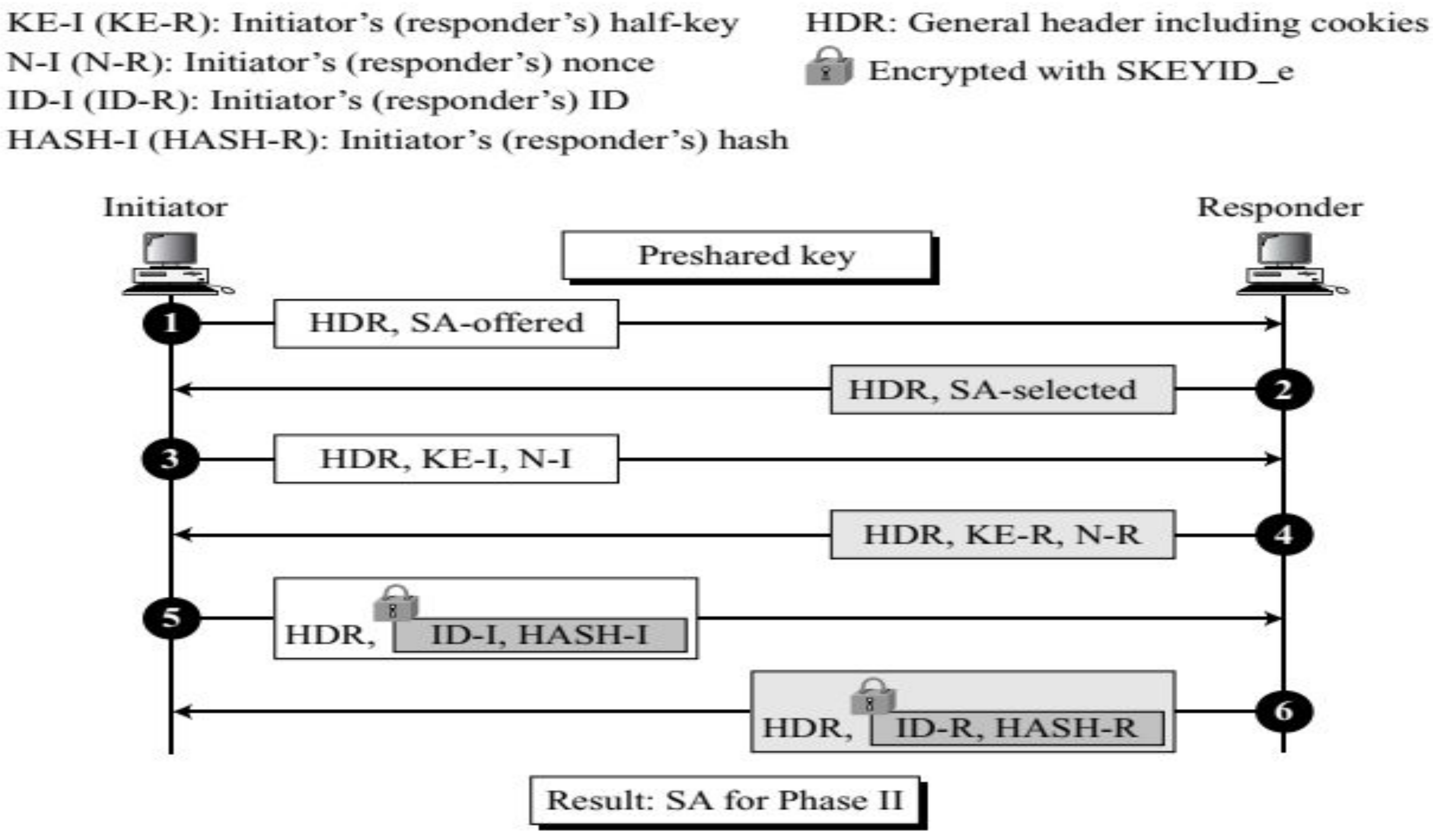
$SKEYID\_a = \boldsymbol{prf}\,(\text{SKEYID, SKEYID\_d} \mid g^{ir} \mid \text{Cookie-I} \mid \text{Cookie-R} \mid 1)$

$SKEYID\_e = \boldsymbol{prf}\,(\text{SKEYID, SKEYID\_a} \mid g^{ir} \mid \text{Cookie-I} \mid \text{Cookie-R} \mid 2)$

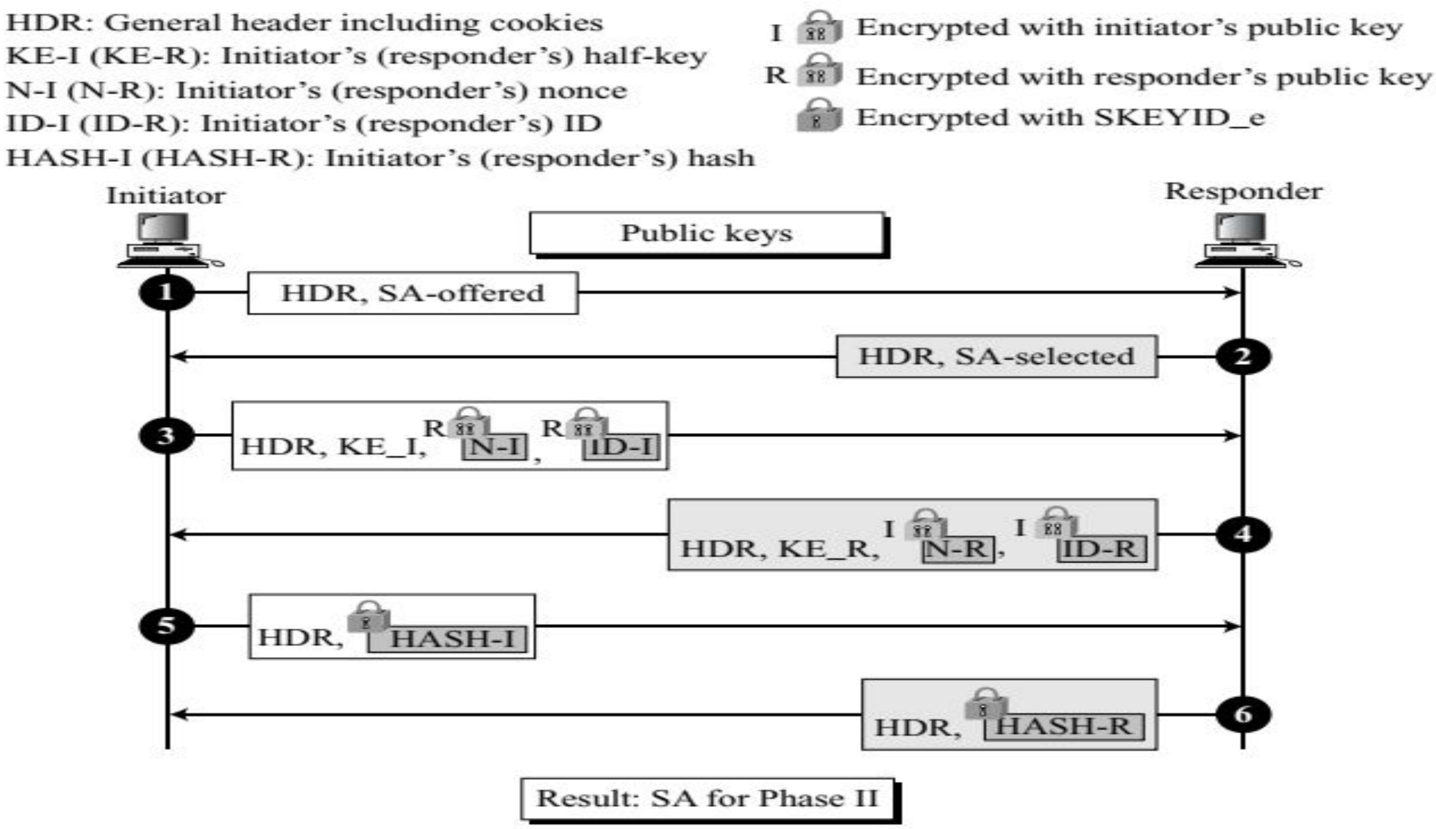
$\text{HASH-I} = \boldsymbol{prf}\,(\text{SKEYID, KE-I} \mid \text{KE-R} \mid \text{Cookie-I} \mid \text{Cookie-R} \mid \text{SA-I} \mid \text{ID-I})$

$\text{HASH-R} = \boldsymbol{prf}\,(\text{SKEYID, KE-I} \mid \text{KE-R} \mid \text{Cookie-I} \mid \text{Cookie-R} \mid \text{SA-I} \mid \text{ID-R})$

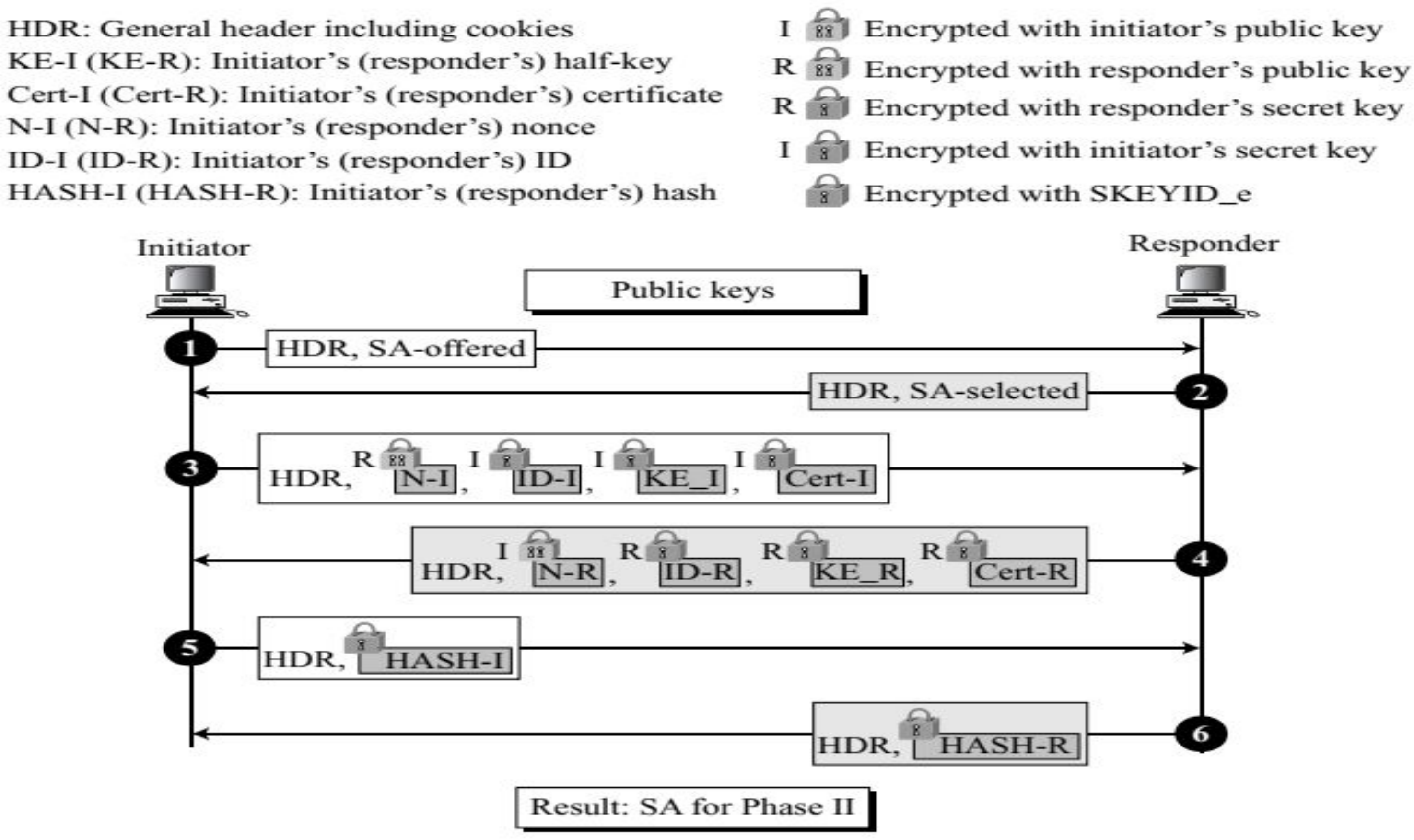
Preshared Secret-Key Method



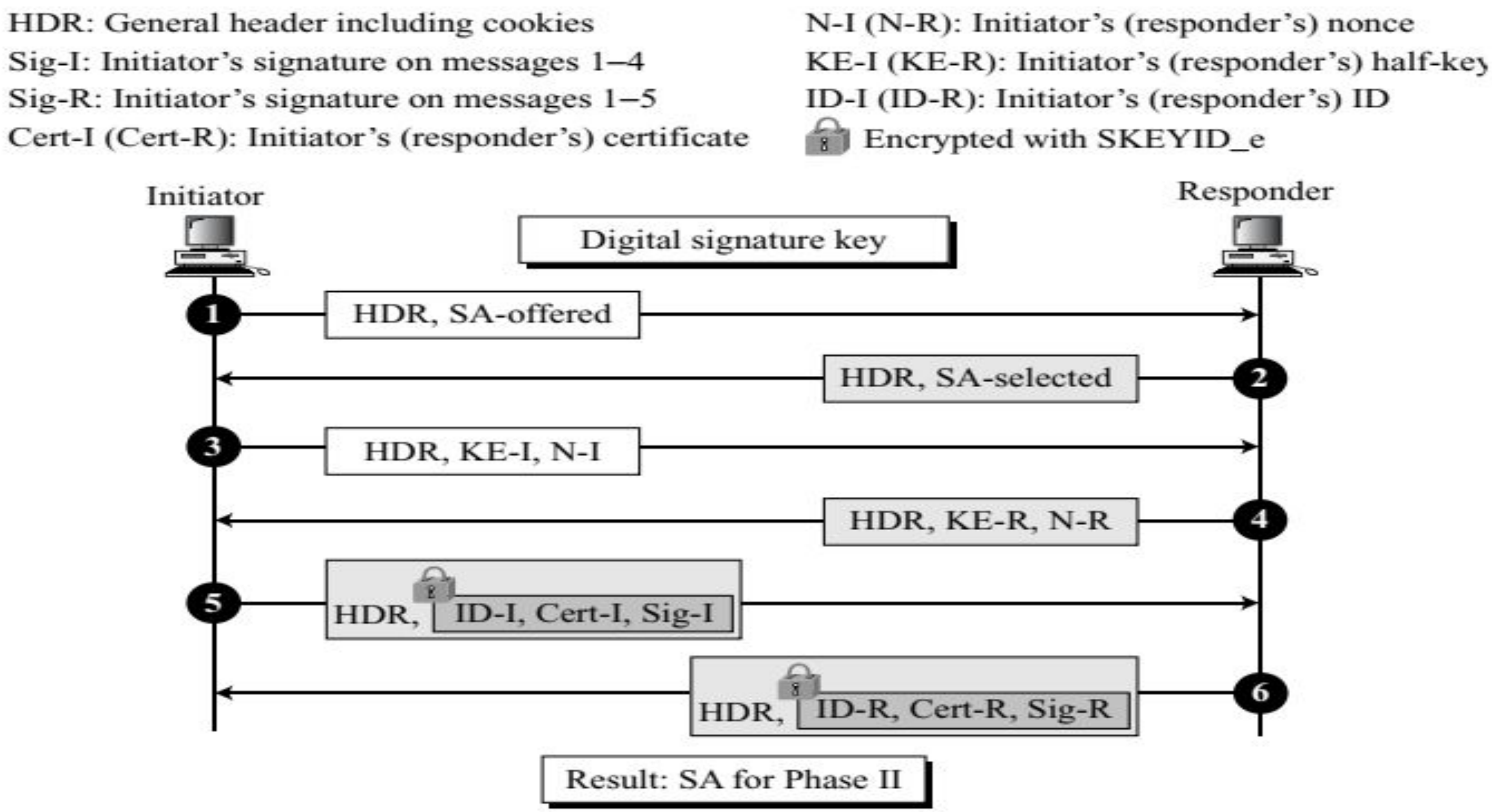
Original Public-Key Method



Revised Public-Key Method



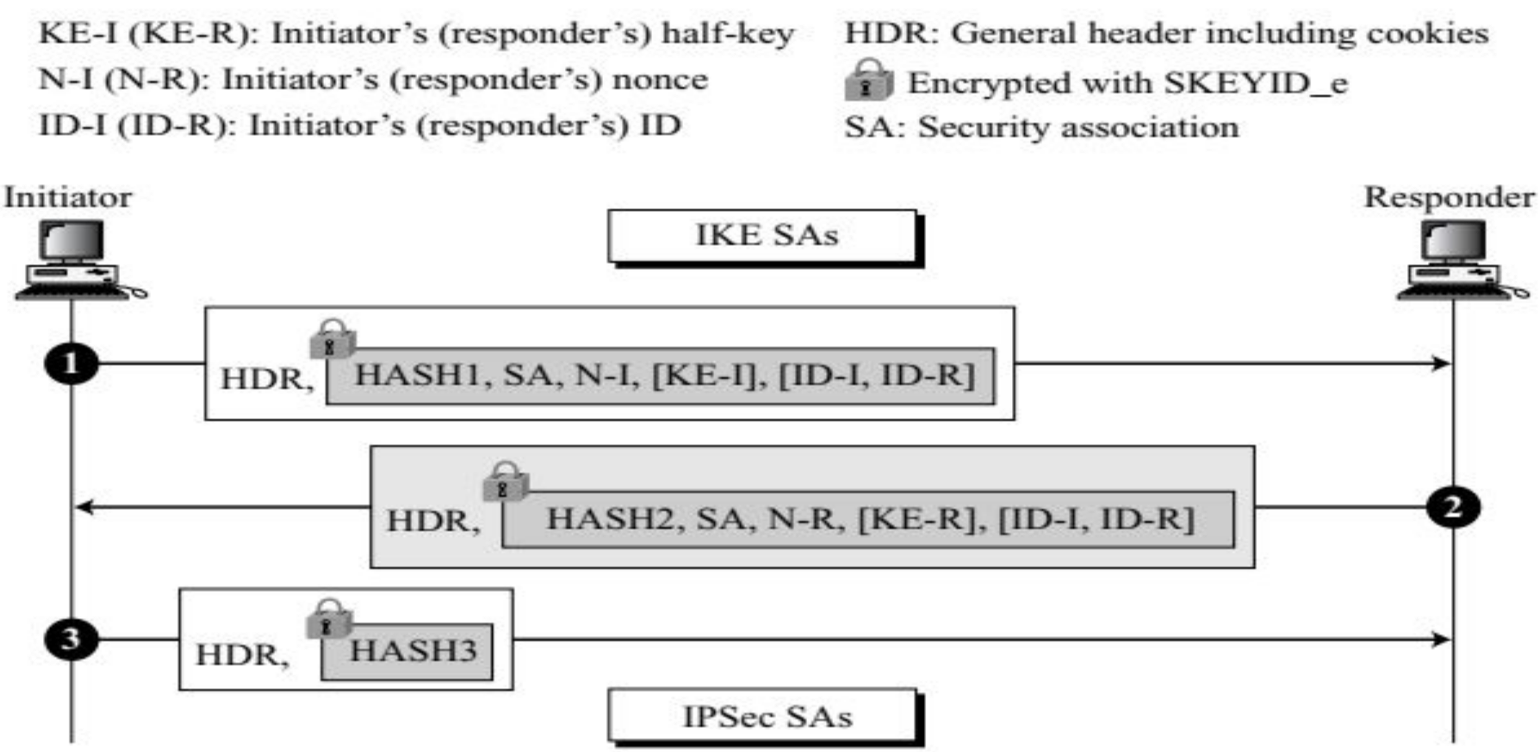
Digital Signature Method



Phase I: Aggressive Mode

- Each aggressive mode is a compressed version of the corresponding main mode.
- Instead of six messages, only three are exchanged. Messages 1 and 3 are combined to make the first message.
- Messages 2, 4, and 6 are combined to make the second message.
- Message 5 is sent as the third message. The idea is the same.

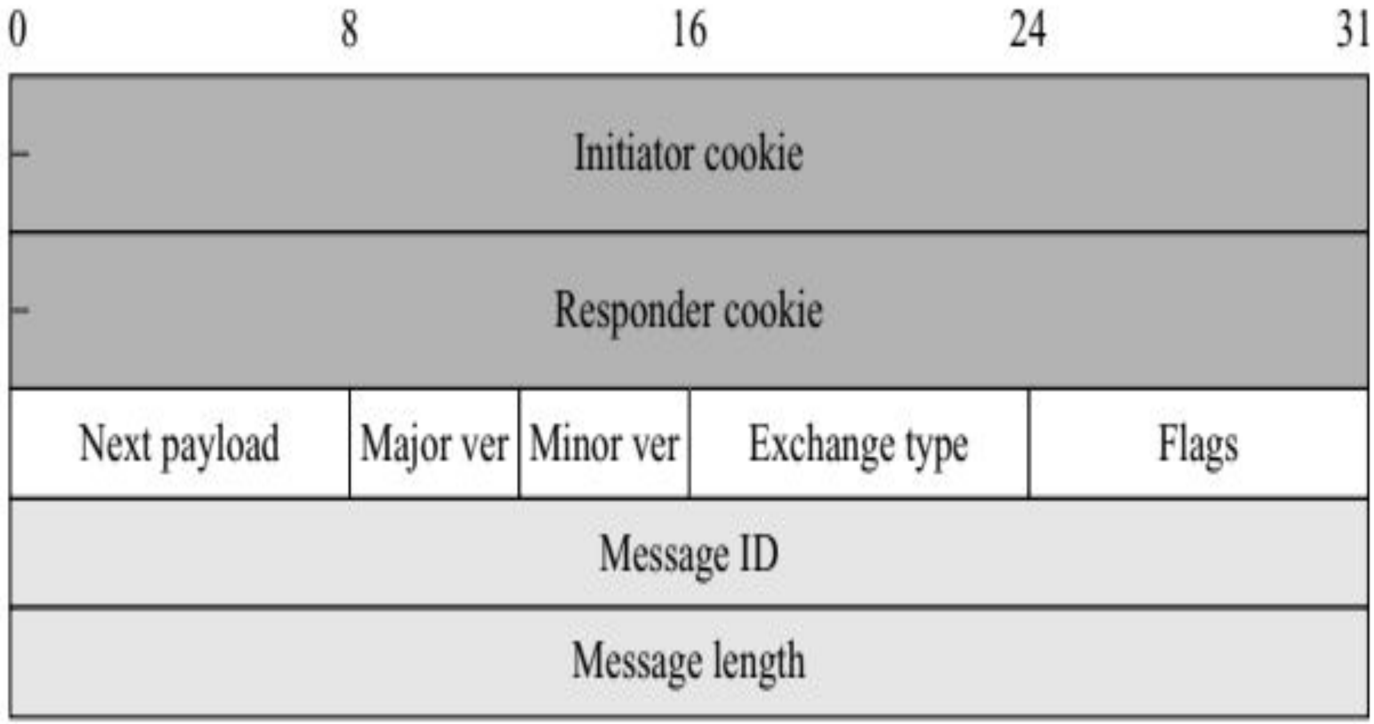
Phase II: Quick Mode



Perfect Forward Security (PFS)

- After establishing an IKE SA and calculating SKEYID_d in phase I, all keys for the quick mode are derived from SKEYID_d.
- Since multiple phase II can be derived from a single phase I, phase II security is at risk if the intruder has access to SKEYID_d.
- To prevent this from happening, IKE allows Perfect Forward Security (PFS) as an option.
- In this option, an additional Diffie-Hellman half-key is exchanged and the resulting shared key (gir) is used in the calculation of key material for IPSec.
- PFS is effective if the Diffie-Hellman key is immediately deleted after the calculation of the key material for each quick mode.

ISAKMP General Header



Types	*Name*	*Brief Description*
0	None	Used to show the end of the payloads
1	SA	Used for starting the negotiation
2	Proposal	Contains information used during SA negotiation
3	Transform	Defines a security transform to create a secure channel
4	Key Exchange	Carries data used for generating keys
5	Identification	Carries the identification of communication peers
6	Certification	Carries a public-key certificate
7	Certification Request	Used to request a certificate from the other party
8	Hash	Carries data generated by a hash function
9	Signature	Carries data generated by a signature function
10	Nonce	Carries randomly generated data as a nonce
11	Notification	Carries error message or status associated with an SA
12	Delete	Carries one more SA that the sender has deleted
13	Vendor	Defines vendor-specification extensions

Generic payload header

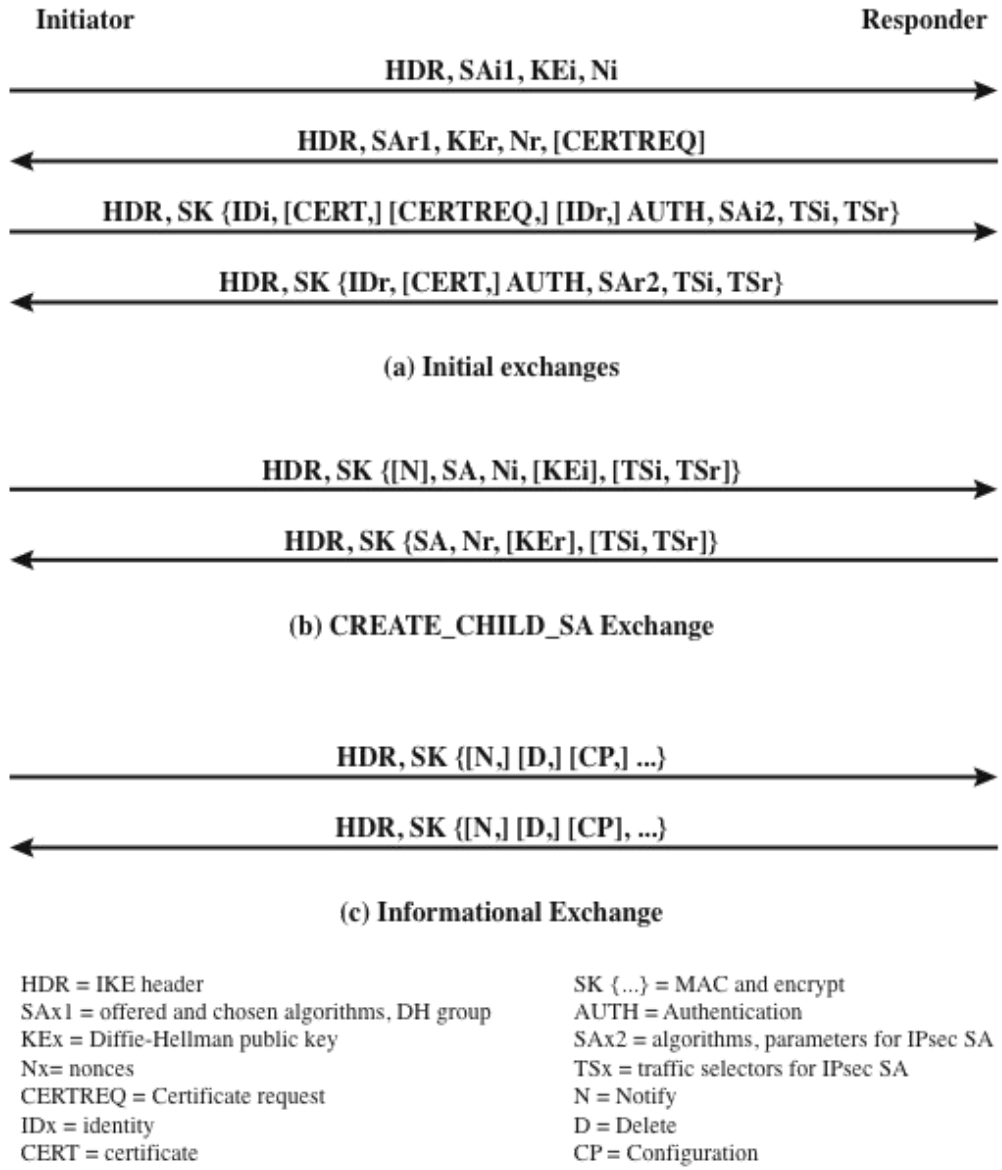
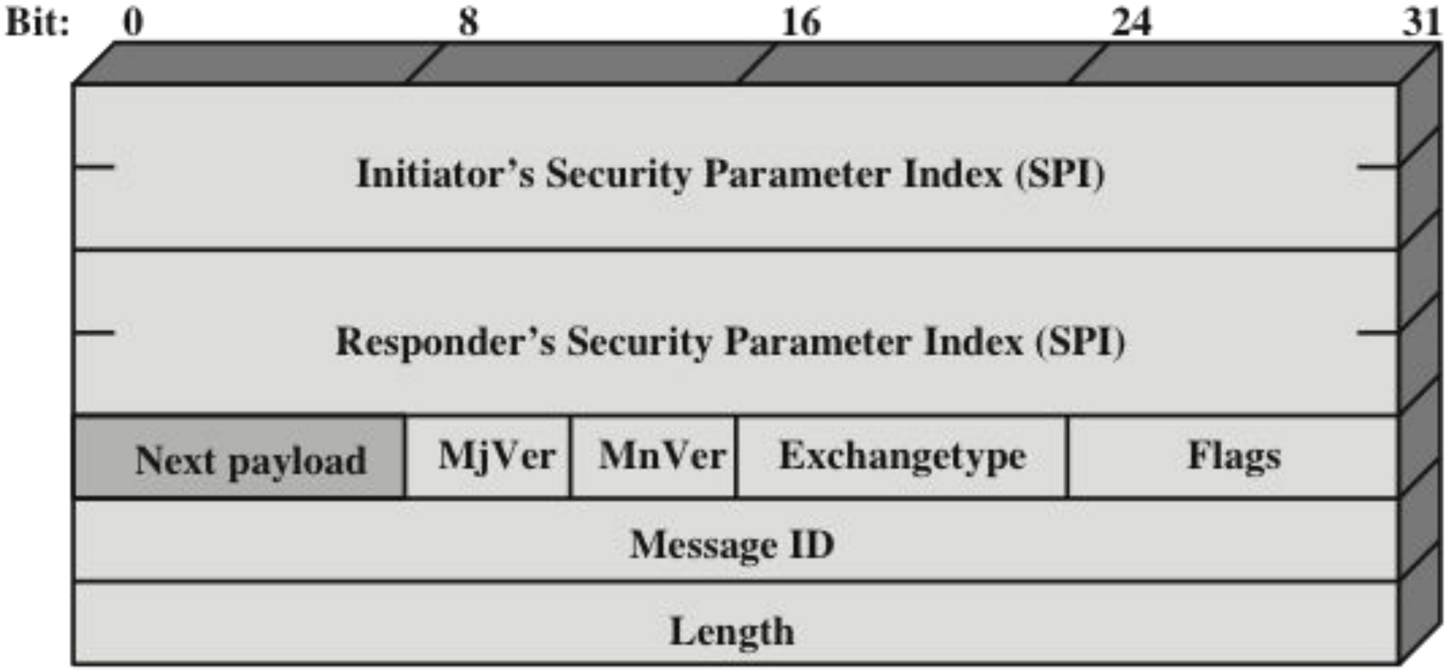


Figure 20.11 IKEv2 Exchanges



(a) IKE Header

(b) Generic Payload Header

Figure 20.12 IKE Formats

IKE Payload Types

Type	Parameters
Security Association	Proposals
Key Exchange	DH Group #, Key Exchange Data
Identification	ID Type, ID Data
Certificate	Cert Encoding, Certificate Data
Certificate Request	Cert Encoding, Certification Authority
Authentication	Auth Method, Authentication Data
Nonce	Nonce Data
Notify	Protocol-ID, SPI Size, Notify Message Type, SPI, Notification Data
Delete	Protocol-ID, SPI Size, # of SPIs, SPI (one or more)
Vendor ID	Vendor ID
Traffic Selector	Number of TSs, Traffic Selectors
Encrypted	IV, Encrypted IKE payloads, Padding, Pad Length, ICV
Configuration	CFG Type, Configuration Attributes
Extensible Authentication Protocol	EAP Message

Table 20.4 Cryptographic Suites for IPsec

(a) Virtual private networks (RFC 4308)

	VPN-A	VPN-B
ESP encryption	3DES-CBC	AES-CBC (128-bit key)
ESP integrity	HMAC-SHA1-96	AES-XCBC-MAC-96
IKE encryption	3DES-CBC	AES-CBC (128-bit key)
IKE PRF	HMAC-SHA1	AES-XCBC-PRF-128
IKE Integrity	HMAC-SHA1-96	AES-XCBC-MAC-96
IKE DH group	1024-bit MODP	2048-bit MODP

(b) NSA Suite B (RFC 4869)

	GCM-128	GCM-256	GMAC-128	GMAC-256
ESP encryption/ Integrity	AES-GCM (128-bit key)	AES-GCM (256-bit key)	Null	Null
ESP integrity	Null	Null	AES-GMAC (128-bit key)	AES-GMAC (256-bit key)
IKE encryption	AES-CBC (128-bit key)	AES-CBC (256-bit key)	AES-CBC (128-bit key)	AES-CBC (256-bit key)
IKE PRF	HMAC-SHA-256	HMAC-SHA-384	HMAC-SHA-256	HMAC-SHA-384
IKE Integrity	HMAC-SHA-256-128	HMAC-SHA-384-192	HMAC-SHA-256-128	HMAC-SHA-384-192
IKE DH group	256-bit random ECP	384-bit random ECP	256-bit random ECP	384-bit random ECP

Summary

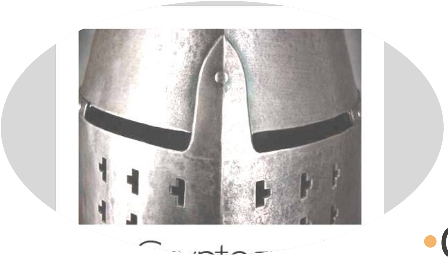
IP security overview

- Applications of IPsec
- Benefits of IPsec
- Routing applications
- IPsec documents
- IPsec services
- Transport and tunnel modes

IP security policy

- Security associations
- Security association database
- Security policy database
- IP traffic processing

- Cryptographic suites



- Encapsulating security payload
  - ESP format
  - Encryption and authentication algorithms
  - Padding
  - Anti-replay service
  - Transport and tunnel modes
- Combining security associations
  - Authentication plus confidentiality
  - Basic combinations of security associations
- Internet key exchange
  - Key determination protocol
  - Header and payload formats