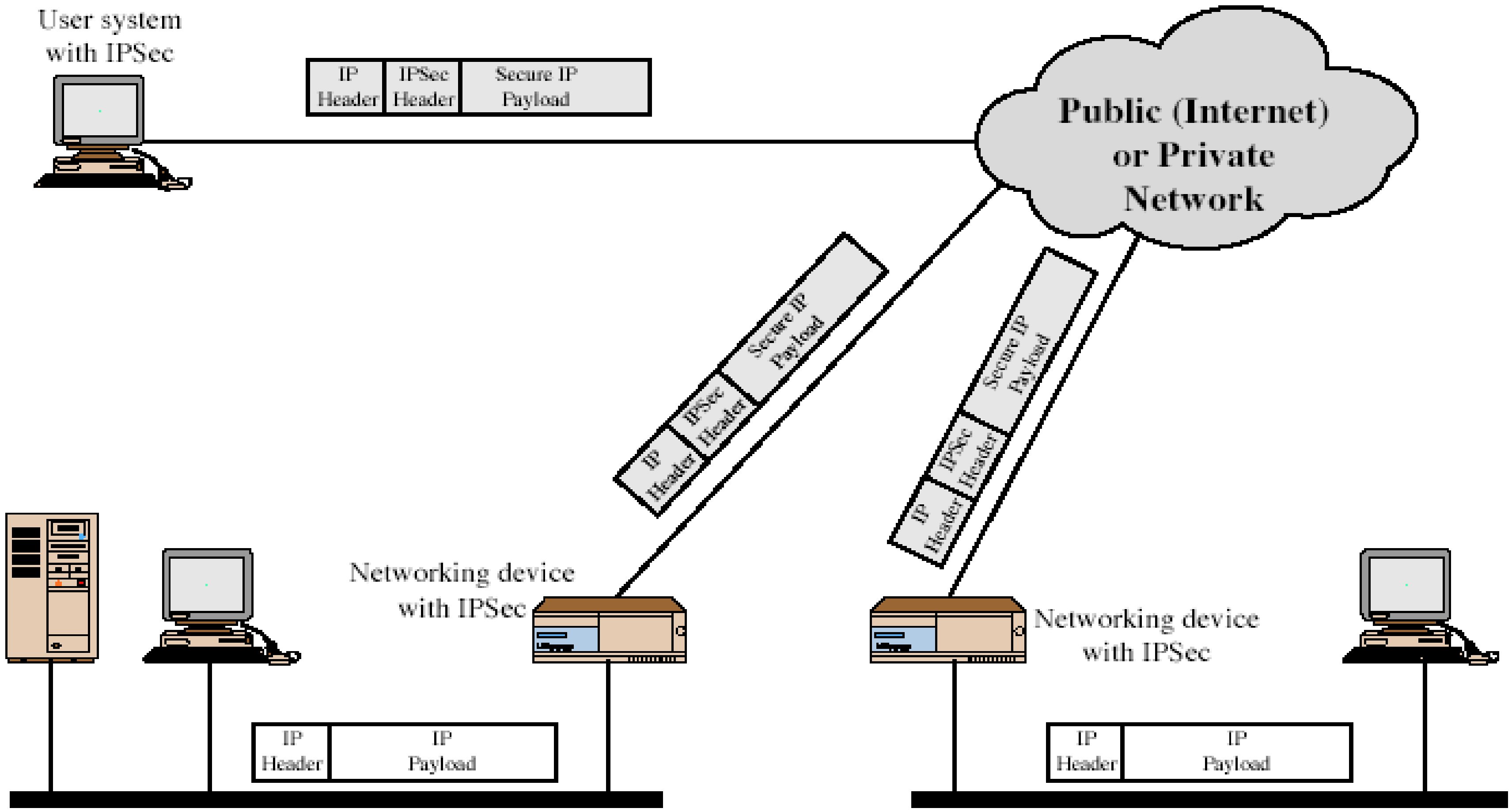
Security for IP layer

- What should it be?
  - Encryption (of IP packets)
  - Integrity (of IP packets)
  - Authentication (two ends of a conversation)
- How should it be done?
  - Add new (cryptographic) headers into IP packets
  - Set up the keys and negotiate the algorithms
  - Authentication of two ends

IPSec—IP Security

- Provide encryption and integrity protection to IP packets (and authentication of two peers).
  - AH (Authentication Header)
    - An additional header, provides integrity protection
  - ESP (Encapsulating Security Payload)
    - Also an addition header, provides encryption and integrity protection
  - IKE (Internet Key Exchange)
    - Establishing session keys (used for AH & ESP) as well as authentication.
  - Both AH and ESP are called IPSec Headers.
  - Authentication: users and data.

IPSec Scenario



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
- can provide security for individual users if desired

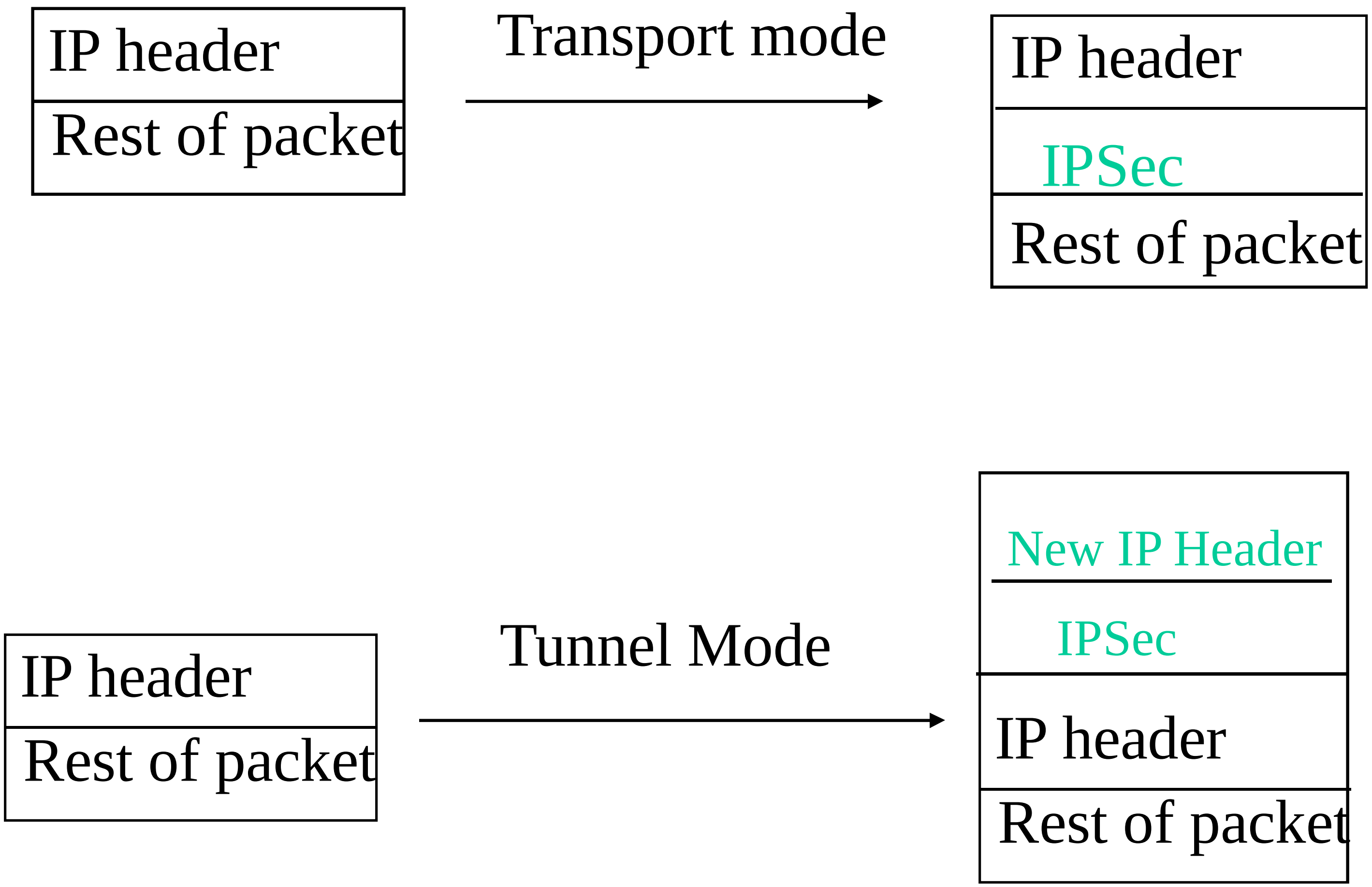
Security Associations (SA)

- A SA is a cryptographically protected connection
  - Two ends (from one end → the other end)
  - For each end:
    - A key
    - cryptographic services being used: e.g., integrity-only, encryption+integrity
    - Cryptographic algorithms
    - identity of the other end,
    - sequence number currently used
- Unidirectional, so a conversation between Alice and Bob consists of two SAs.

SA management

- SA database: store all its SAs.
- Each SA has a related index in SA database, called SPI (Security Parameter Index).
- By IKE, two ends negotiate SA and put SA into their SA database.
- In the IPSec header, there is a SPI field.
- A sender will put SPI in the IPSec's SPI field of every IP packet
- The receiver will look up its SA database to find the SA corresponding to the SPI in the IP packet.

Two modes of IPSec



IP header

Rest of packet

Transport mode

IP header

IPSec

Rest of packet

IP header

Rest of packet

Tunnel Mode

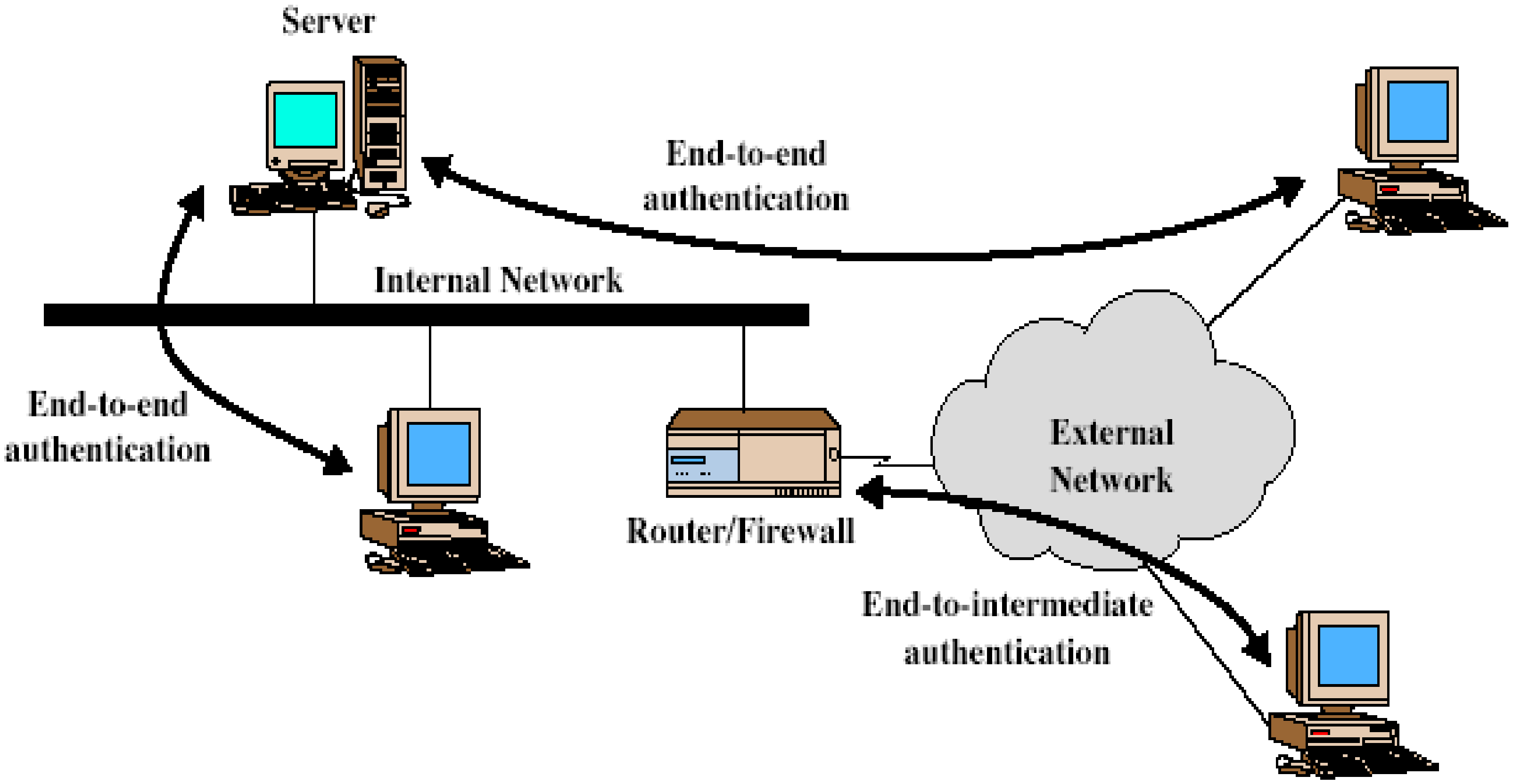
New IP Header

IPSec

IP header

Rest of packet

Transport & Tunnel Modes


Server
End-to-end authentication
Internal Network
End-to-end authentication
Router/Firewall
External Network
End-to-intermediate authentication

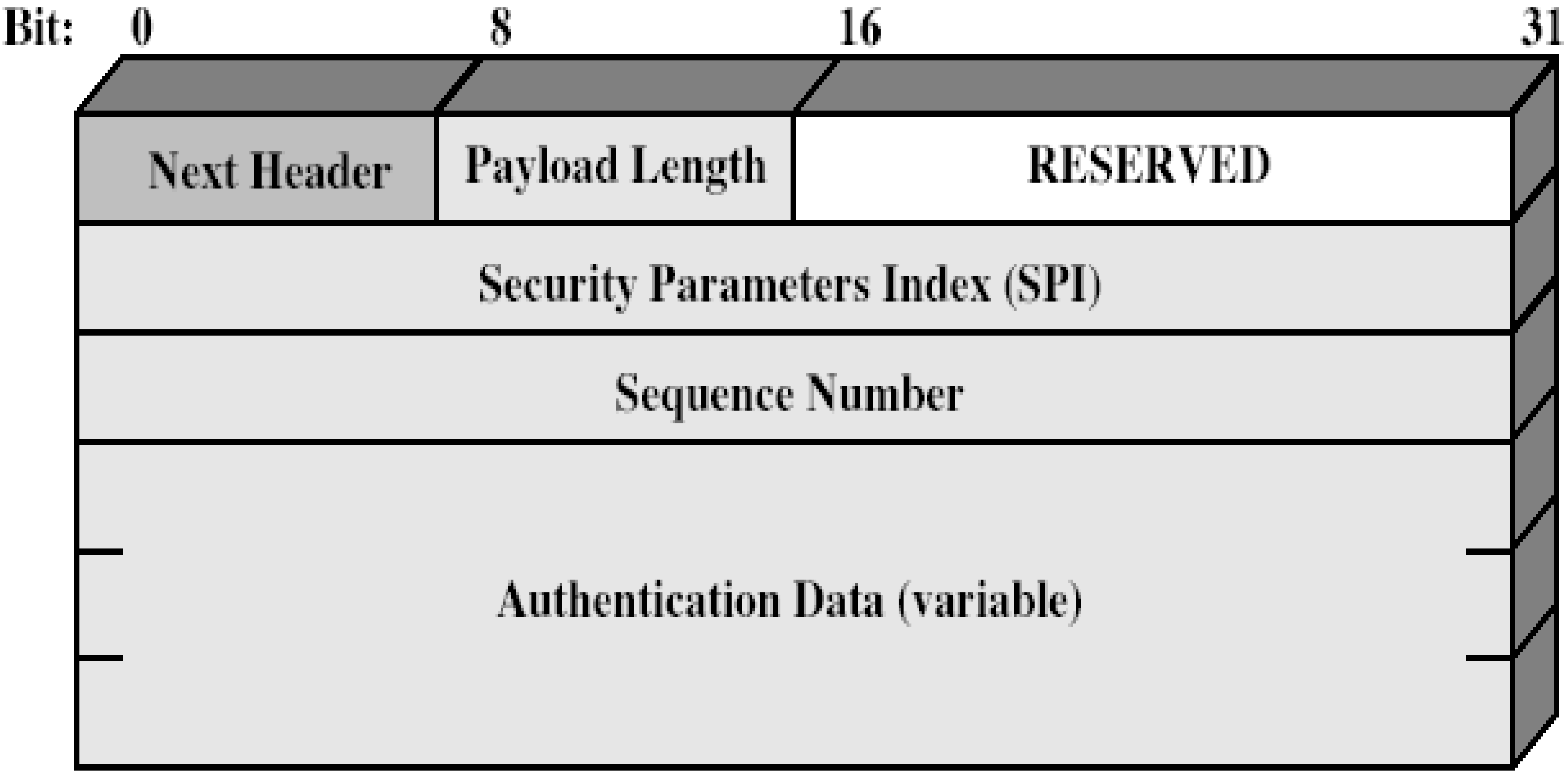
IPSec mode usage

- Transport mode is used when IPSec is used end-to-end
- Tunnel mode is used between firewalls or endnode and firewall. (Example)
- Combination of multiple modes
- In tunnel mode, the original IP packet will be kept intact (not really?).

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



Bit: 0 8 16 31

Next Header | Payload Length | RESERVED

Security Parameters Index (SPI)

Sequence Number

Authentication Data (variable)

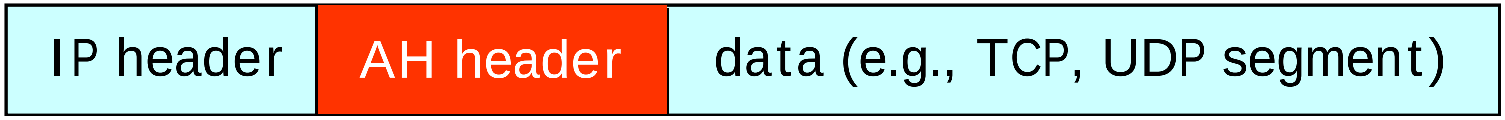
Authentication Header (AH)

#Octets	Field Name
1	type of next header (IP: 4, TCP: 6, UDP:17, ESP: 50, AH: 51)
1	length of the AH header (in unit of 32/64 bits)
2	unused
4	SPI (Security Parameter Index)
4	Sequence number of this AH packet
variable	Authentication data

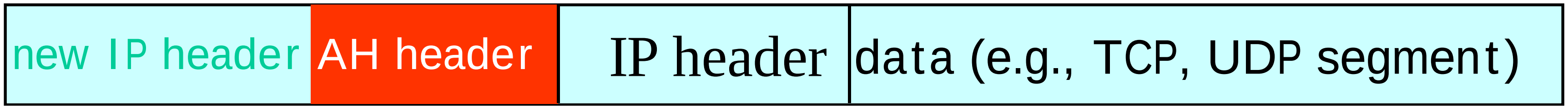
Note: authentication data is the MAC for
the immutable fields in IP header + data of IP packet

Mutable field: TTL, header checksum, type of service, Flags, Fragment offset (IPv4)
Hop Limit, Type Of service, Flow Label, … (IPv6)

AH Protocol (cont.)



IP header | AH header | data (e.g., TCP, UDP segment)

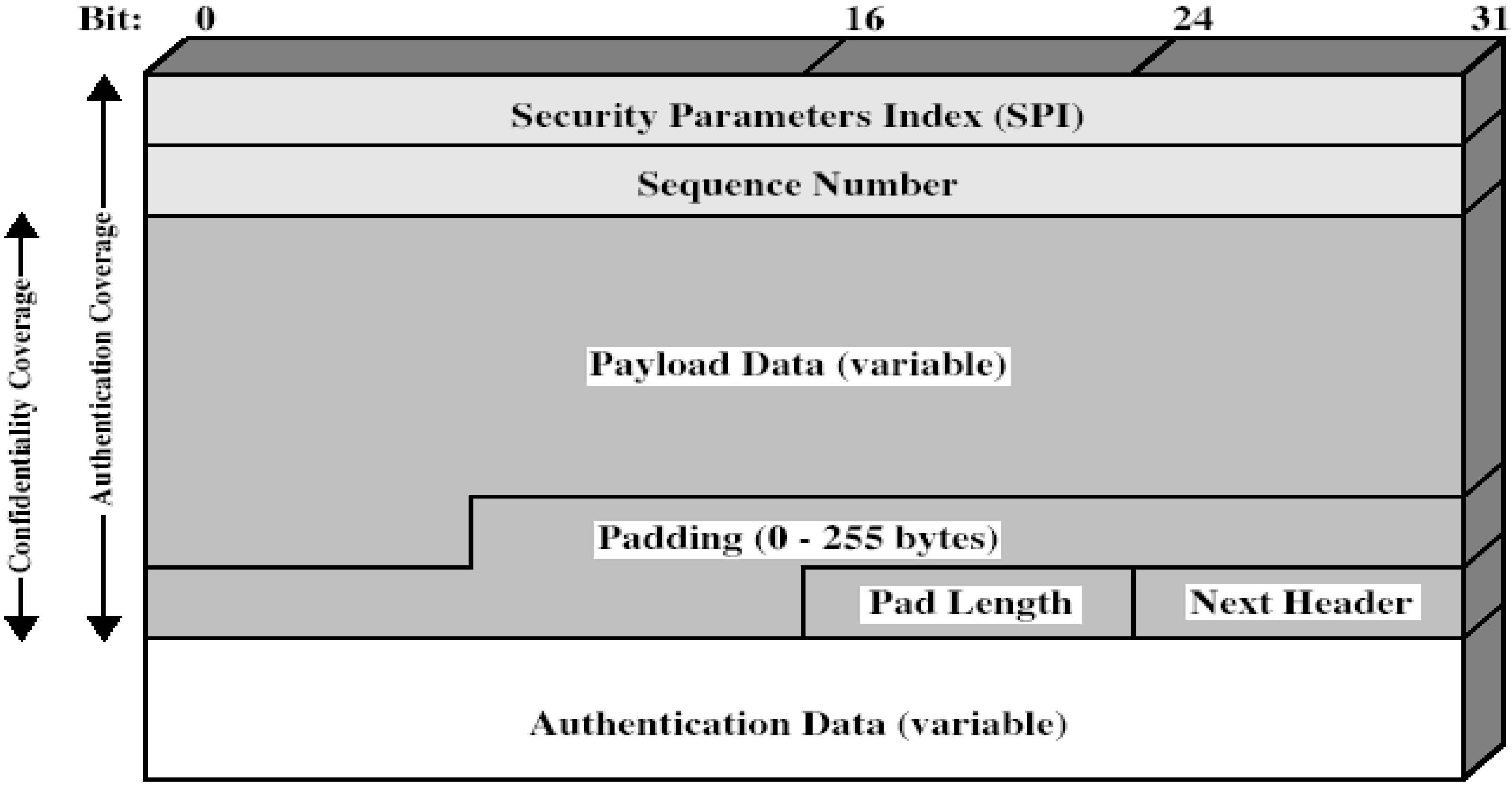


new IP header | AH header | IP header | data (e.g., TCP, UDP segment)

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



Bit: 0 16 24 31

Security Parameters Index (SPI)

Sequence Number

Payload Data (variable)

Padding (0 - 255 bytes)

Pad Length

Next Header

Authentication Data (variable)

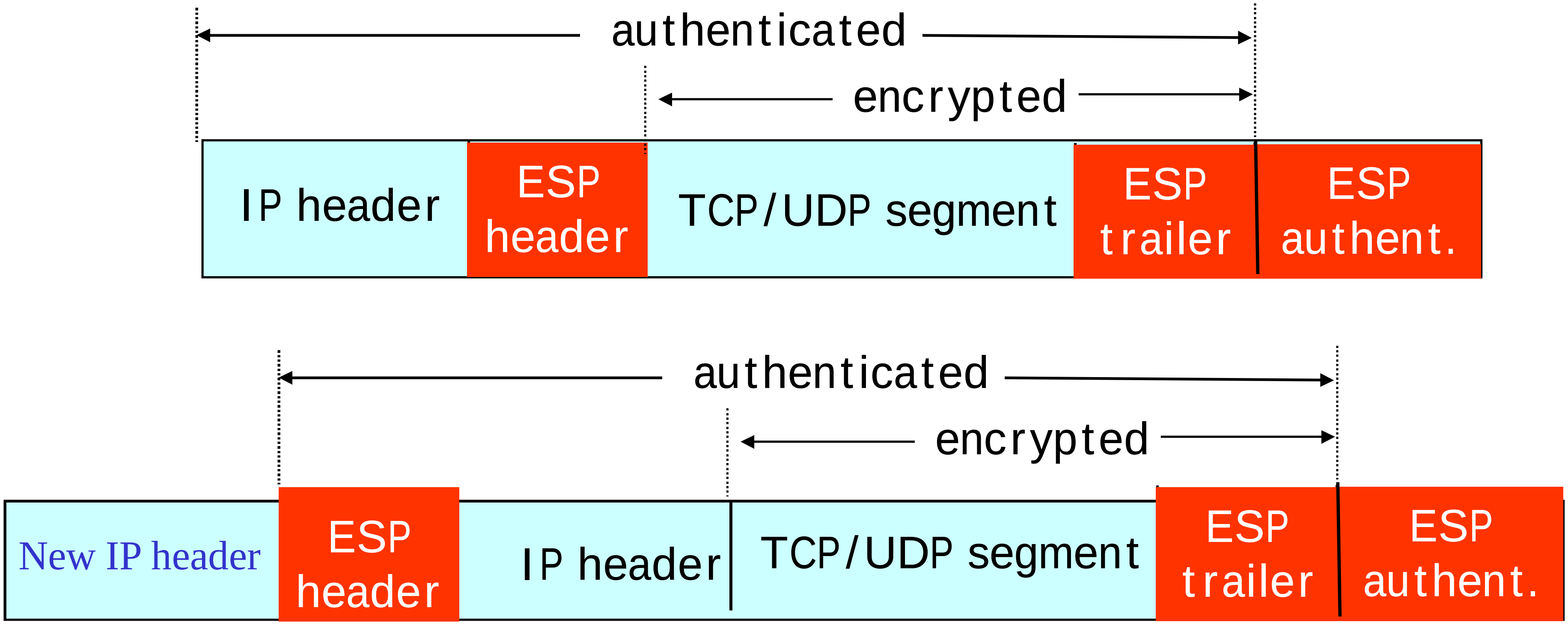
Confidentiality Coverage

Authentication Coverage

Encapsulating Security Payload (ESP)

#octets	field name
4	SPI
4	Sequence number
variable	IV
variable	Data (Original IP payload)
variable	padding
1	Padding length (in units of octets)
1	Next header/protocol type
variable	Authentication data

ESP Protocol (Cont.)



ESP in fact puts information both before and after the protected data.
For encryption, DATA, padding, padding length and next header are encrypted.
For authentication, all fields are included.

AH vs. ESP

- AH just does integrity
- ESP does both encryption & integrity
- Therefore
  - If just integrity, use AH or ESP
  - If both integrity and encryption, then use both AH and ESP, or just use ESP.
  - ESP always does encryption, so if just integrity, the ESP uses a null encryption algorithm.

IKE—Internet key exchange

- To authenticate each other and negotiate session key used for IPSec.
- IKE history
  - Initial two candidates for IKE
    - Photuris – using Diffie-Hellman key exchange (with signature)
    - SKIP (Simple Key Management for Internet Protocol) – using long term Diffie-Hellman key exchange (i.e., g^a mod p is publicly known)
  - Two fight with each other, no one selected

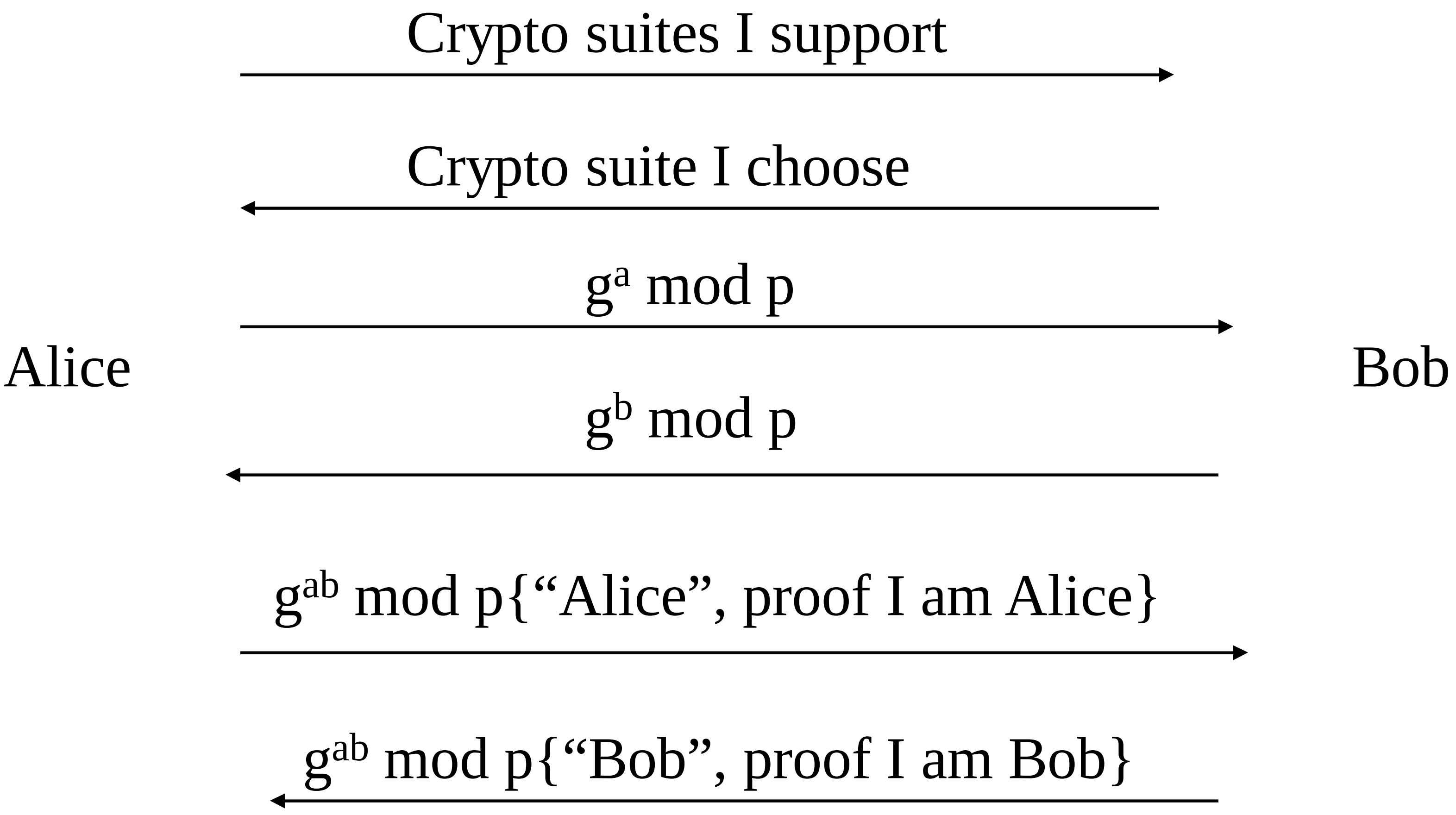
IKE--history

- ISAMKP (Internet Security Association and Key Management Protocol):
  - Provides a framework for authentication and key exchange, (not a protocol, suppose support different key exchange)
- OAKLEY
  - Describe a series of key exchanges and services (e.g. perfect forward secrecy, identity protection, and authentication)
- SKEME (Secure Key Exchange MEchanism)
  - describes a versatile key exchange technique which provides anonymity, repudiability, and quick key refreshment.
- IKE:
  - a hybrid of the ISAKMP framework and the Oakley and SKEME protocols.

IKE phases

- Phase 1
  - Mutual authentication and establishes session keys (used in phase 2) by key exchange, called IKE SA
    - How about authentication:
      - Pre-shared secret key
      - Public encryption key (Know the other end’s public key in advance)
      - Public signature key (send the public key certificate to the other end)
    - Establishes session key
      - Diffie-Hellman key exchange,
      - protected by above keys.
- Phase 2
  - Establish multiple session keys, such as ESP SA, AH SA, …

IKE phase 1—main mode



Crypto suites I support

Crypto suite I choose

g^a mod p

Alice

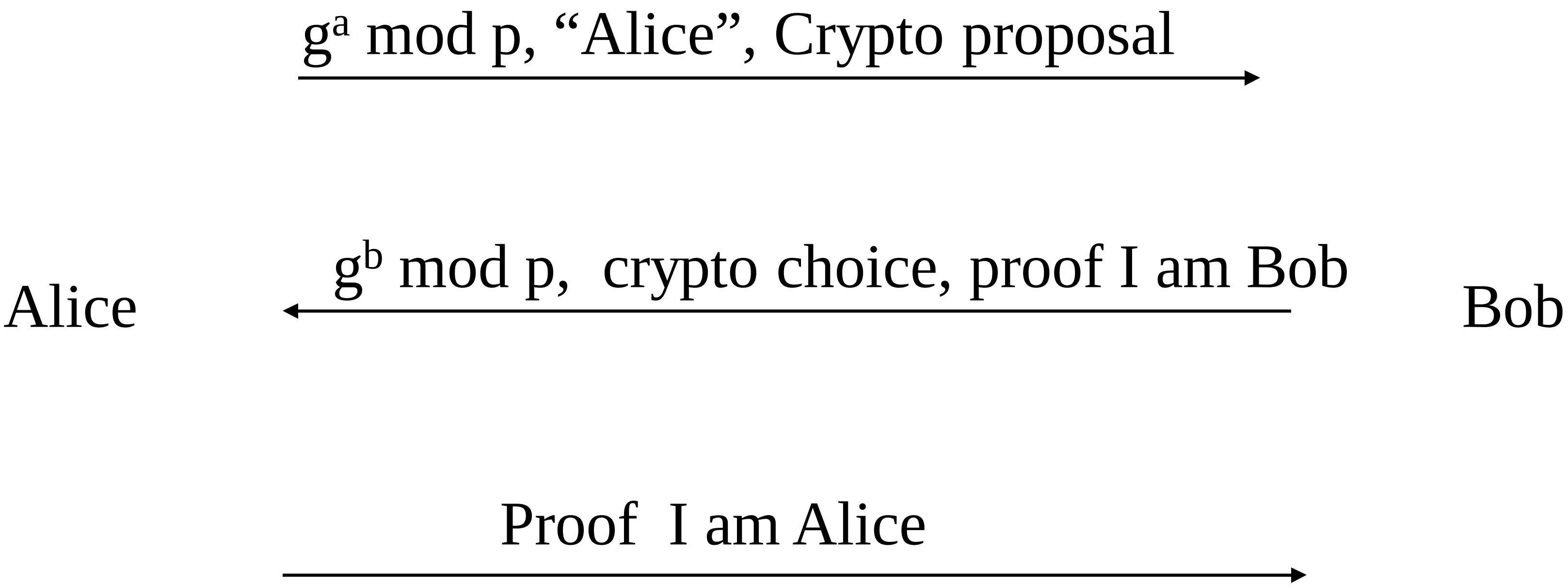
Bob

g^b mod p

g^{ab} mod p{“Alice”, proof I am Alice}

g^{ab} mod p{“Bob”, proof I am Bob}

IKE phase 1—aggressive mode



g^a mod p, “Alice”, Crypto proposal

g^b mod p, crypto choice, proof I am Bob

Alice

Bob

Proof I am Alice

IKE phase 2 –quick mode

- Any party can initiate a quick mode exchange to set up an ESP SA or AH SA
  - Negotiating crypto parameters
  - Optionally doing a Diffie-Hellman exchange (if *perfect forward secrecy* is desired)
  - Negotiating what traffic will be sent on the SA

IKE phase 1: total eight modes

- Two modes
  - Main mode
  - aggressive mode
- Keys
  - Public signature keys
  - Public encryption key, encrypt fields separately
  - Public encryption key, encrypt fields together
  - Pre-shared secret key