# IP Security
# IK2218/EP2120

Markus Hidell, mahidell@kth.se
KTH School of ICT
*Based partly on material by Vitaly Shmatikov, Univ. of Texas*

ROYAL INSTITUTE
OF TECHNOLOGY

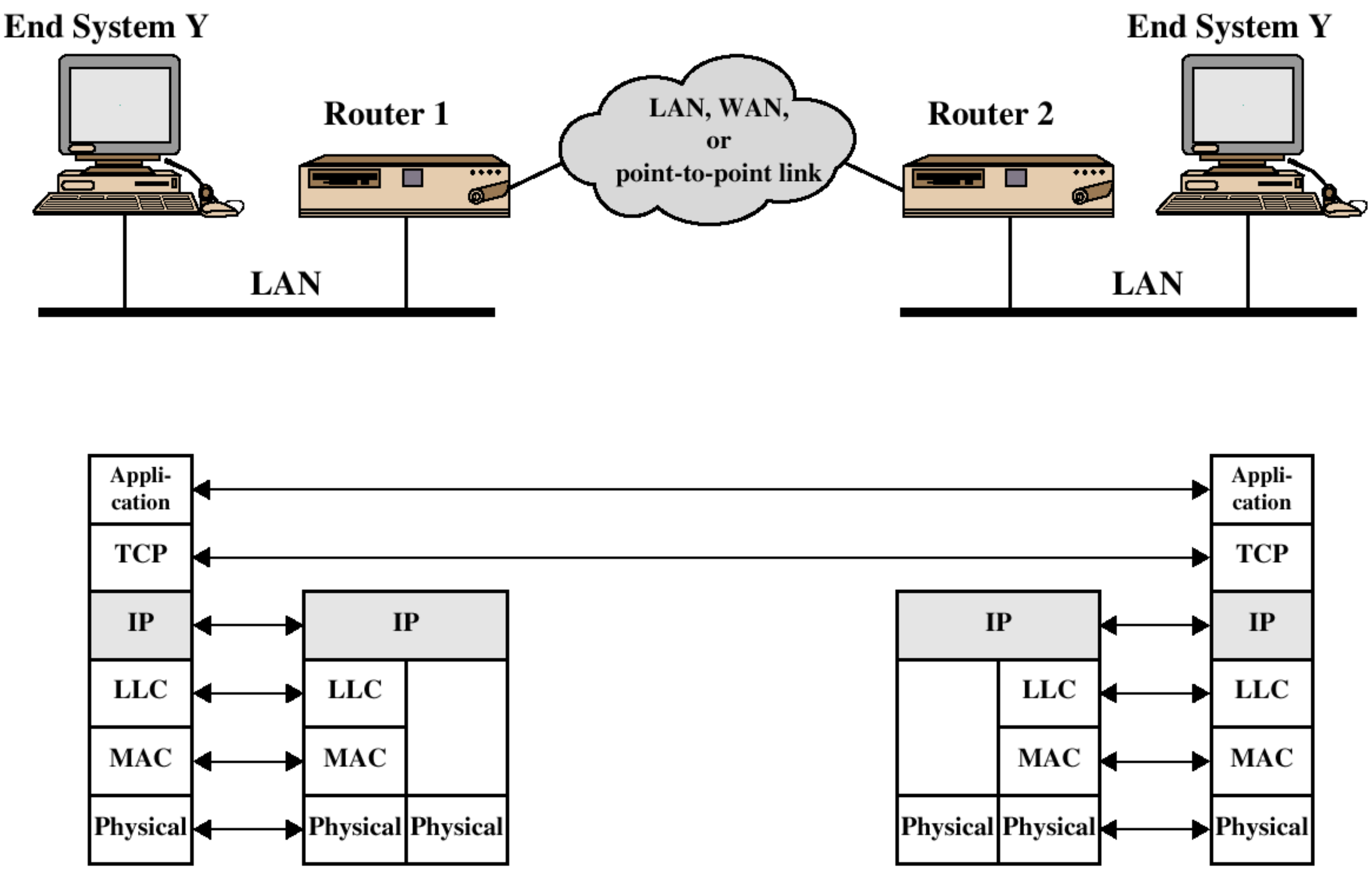
# Acknowledgements


- The presentation builds upon material from
  - Previous slides by Markus Hidell and Peter Sjödin
  - *Computer Networking: A Top Down Approach*, 5th ed. Jim Kurose, Keith Ross. Addison-Wesley.
  - *TCP/IP Protocol Suite*, 4th ed, Behrouz Foruzan. McGraw-Hill.

# Part 1
# IPsec: AH and ESP

Basics, traffic protection

# TCP/IP

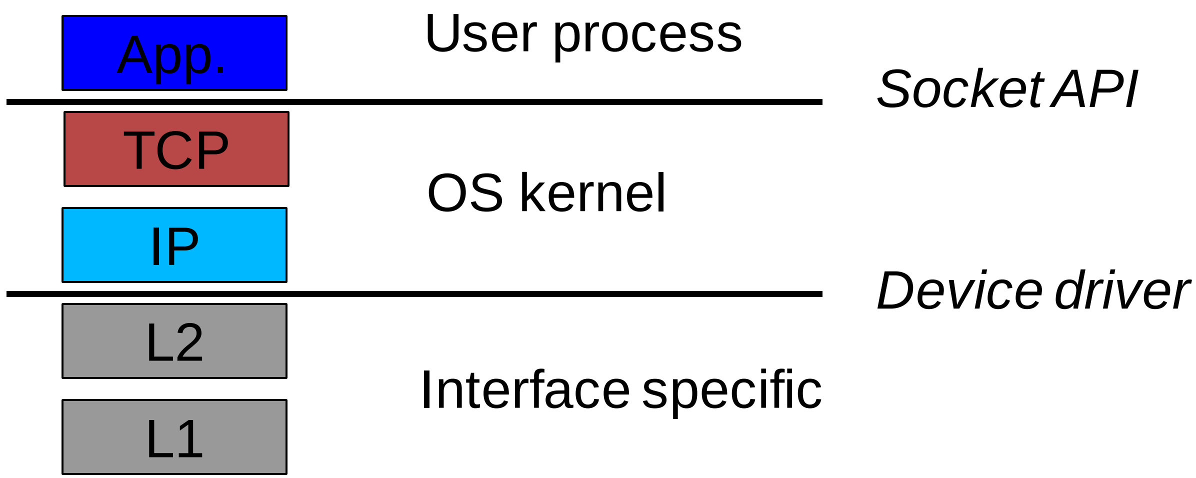

End System Y
Router 1
LAN, WAN,
or
point-to-point link
Router 2
End System Y
LAN
LAN
Appli-
cation
TCP
IP
LLC
MAC
Physical
IP
LLC
MAC
Physical
Physical
IP
LLC
MAC
Physical
Physical
Appli-
cation
TCP
IP
LLC
MAC
Physical

# IP Security Issues

- Eavesdropping
- Modification of packets in transit
- Identity spoofing (forged source IP addresses)
- Denial of service

- Many solutions are application-specific
  - TLS for Web, S/MIME for email, SSH for remote login
- IPsec aims to provide a framework of open standards for secure communications over IP
  - Protect <u>every</u> protocol running on top of IPv4 and IPv6

# Operating System Layers




- SSL (Secure Socket Layer) changes the API to TCP/IP
  - Applications change, but OS doesn't
  - TCP does not participate in the cryptography...(DoS attacks)
- IPsec implemented in OS
  - Applications and API remain unchanged (at least in theory)
- To make full use of IPSec, API and apps have to change!
  - and accordingly also the applications (pass on other IDs than IP addr)

# Overview of IPsec

- Authenticated Keying
  - Internet Key Exchange (IKE)
    - Next part of the lecture
- Data Encapsulation
  - ESP: IP Encapsulating Security Payload (RFC 4303)
  - AH: IP Authentication Header (RFC 4302)
- Security Architecture (RFC 4301)
  - Tunnel/transport Mode
  - Databases (Security Association, Policy, Peer Authorization)

# IPsec: Network Layer Security

IPsec = AH + ESP + IKE

Protection for IP traffic
AH provides integrity and origin authentication
ESP also confidentiality

Sets up keys and algorithms for AH and ESP

- AH and ESP rely on an existing security association
  - Idea: parties must share a set of secret keys and agree on each other's IP addresses and crypto algorithms
- Internet Key Exchange (IKE)
  - Goal: establish security association for AH and ESP
  - If IKE is broken, AH and ESP provide no protection!

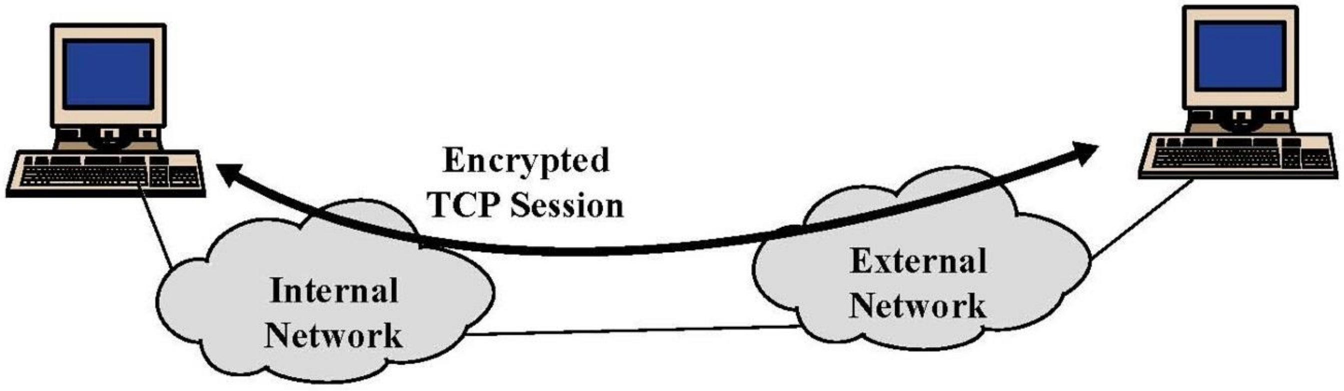
# IPsec Security Services

- Authentication and integrity for packet sources
  - Ensures connectionless integrity (for a single packet) and protection against packet replay (partial sequence integrity)
- Confidentiality (encapsulation) for packet contents
- Authentication and encapsulation can be used separately or together
- Either provided in one of two modes
  - Transport mode
  - Tunnel mode

# IPsec Modes

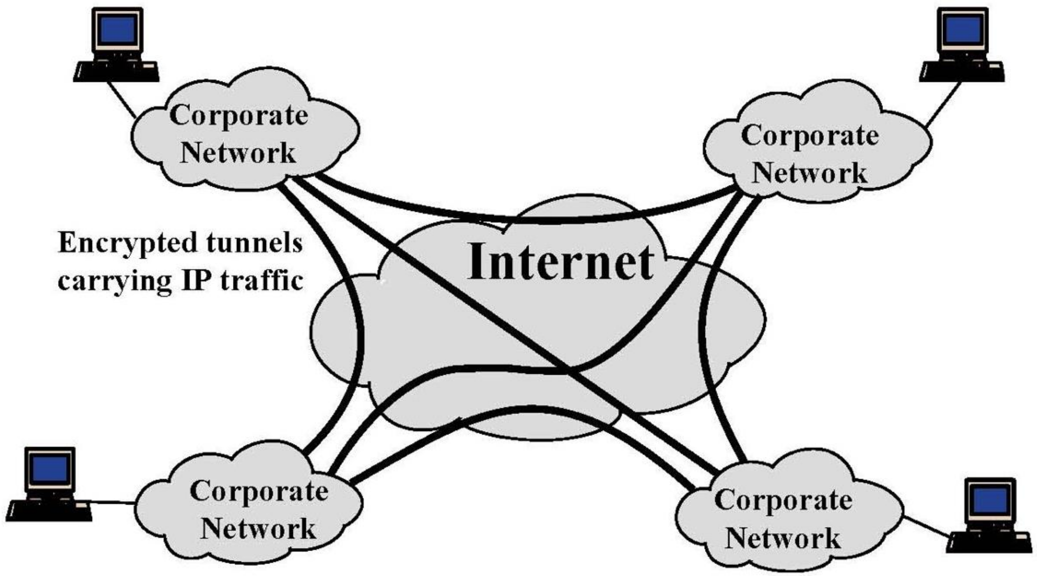
- Transport mode
  - Used to deliver services from host to host or from host to gateway
  - Usually within the same network, but can also be end-to-end across networks
- Tunnel mode
  - Used to deliver services from gateway to gateway or from host to gateway
  - Usually gateways owned by the same organization
    - With an insecure network in the middle

# IPsec in Transport Mode



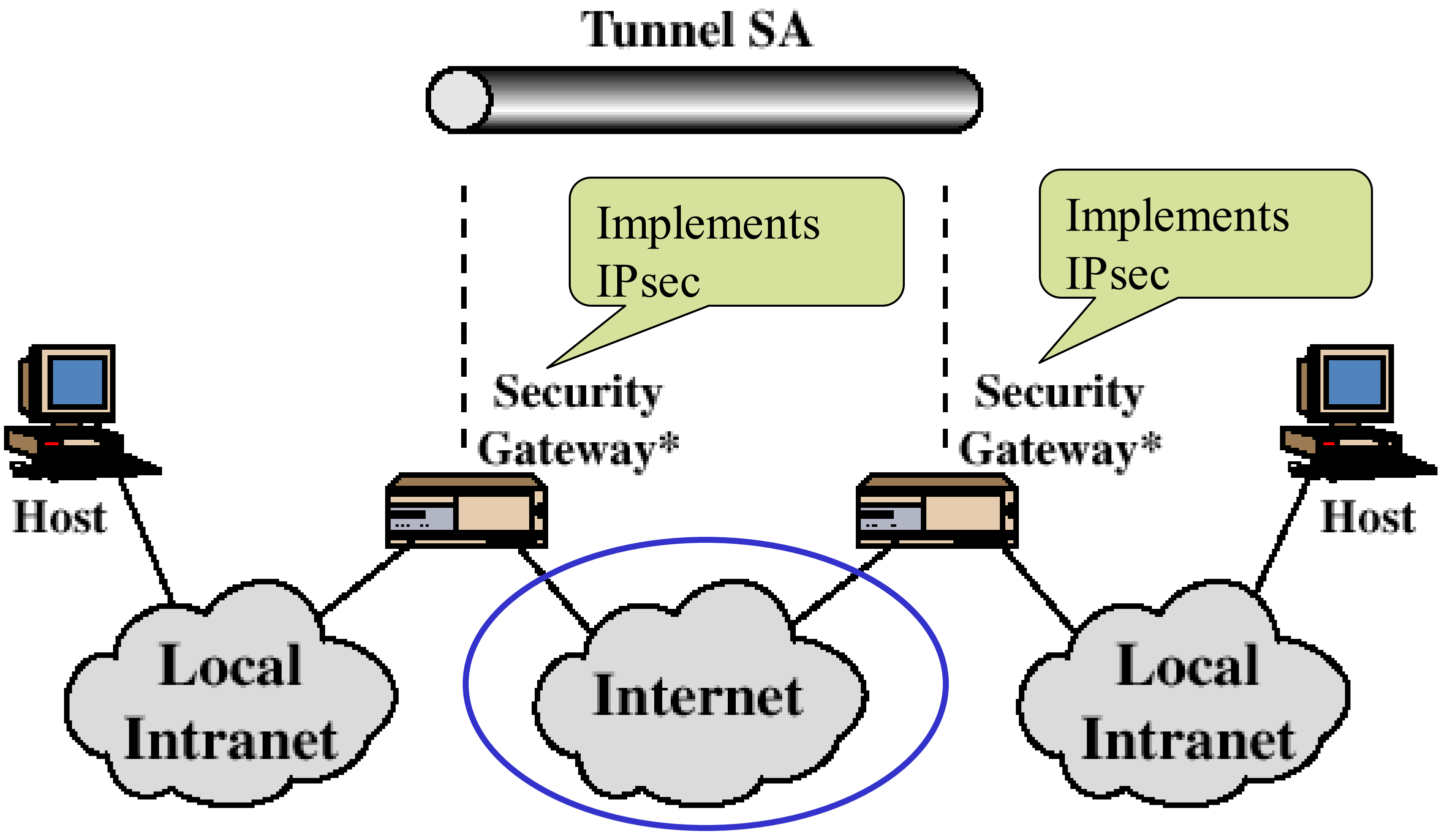

- End-to-end security between two hosts
- Requires IPsec support at each host

# IPsec in Tunnel Mode




- Gateway-to-gateway security
  - Internal traffic behind gateways not protected
  - Typical application: virtual private network (VPN)
- Only requires IPsec support at gateways
  - API /application changes not an issue

# Tunnel Mode Illustration




IPsec protects communication on the insecure part of the network

# Transport Mode vs Tunnel Mode

- Transport mode secures packet payload and leaves IP header unchanged

| | | |
|---|---|---|
| IP header (real dest) | IPsec header | TCP/UDP hdr + data |

- Tunnel mode encapsulates both IP header and payload into IPsec packets

| | | | |
|---|---|---|---|
| IP header (gateway) | IPsec header | IP header (real dest) | TCP/UDP hdr + data |

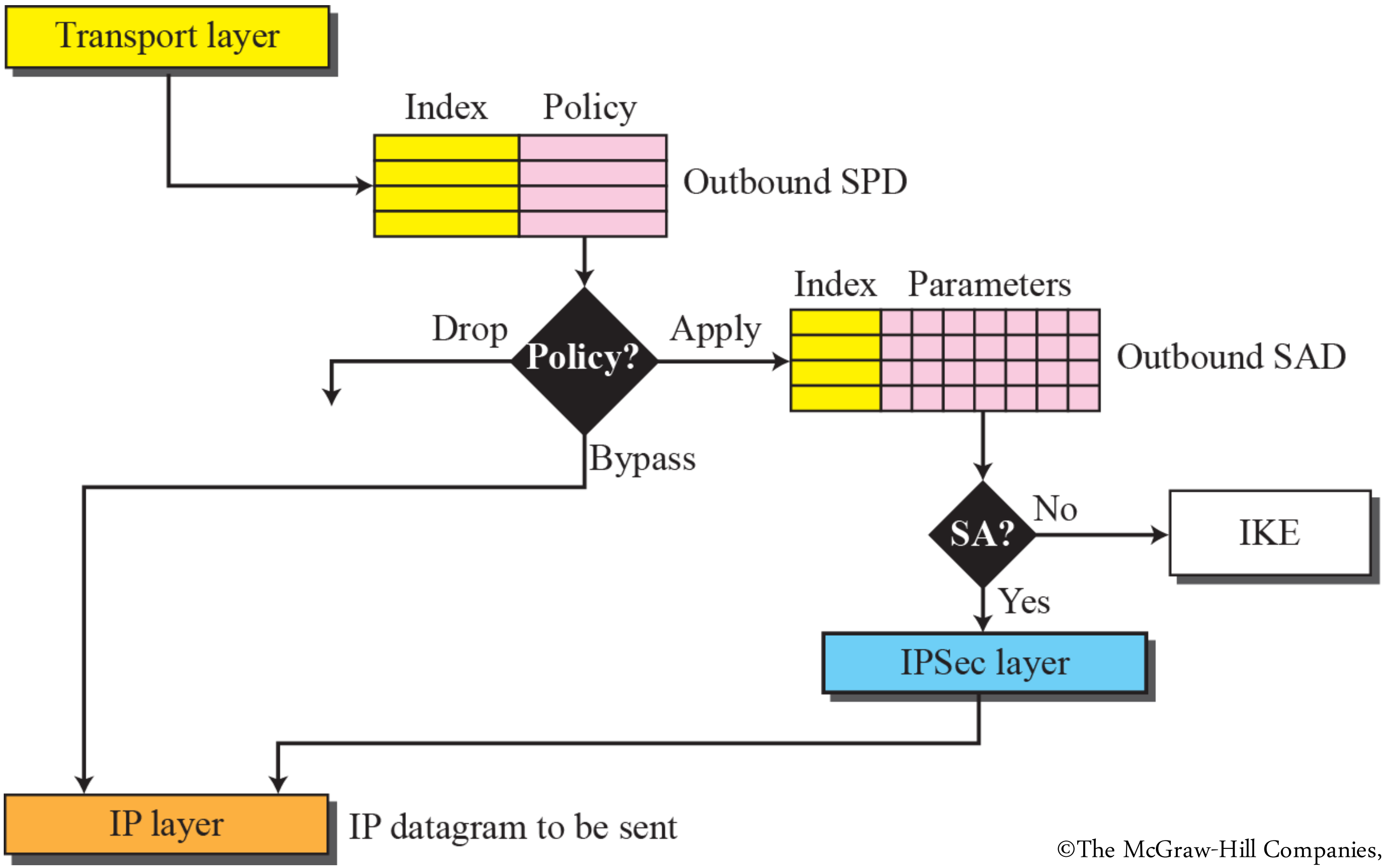
# Security Association (SA)

- One-way sender-recipient relationship
  - Manually configured or negotiated through IKE
- SA determines how packets are processed
  - Cryptographic algorithms, keys, AH/ESP, lifetimes, sequence numbers, mode (transport or tunnel)
- SA is uniquely identified by {SPI, dst IP addr, flag}
  - SPI: Security Parameter Index
    - Chosen by destination (unless traffic is multicast...)
  - Flag: ESP or AH
  - Each IPsec implementation keeps a database of SAs
  - SPI is sent with packet, tells recipient which SA to use

# Sending IPsec Packets

- When Alice is sending to Bob:
- Consult “security policy database” (SPD) to check if packet should be protected with IPsec or not (defined by selectors)
  - SPD can be compared with a firewall table
- SPD provides pointer to the associated SA entry in the security association database (SAD)
- SA provides SPI, algorithm, key, sequence number, etc.
- Include the SPI in the message

# Outbound IPsec Processing



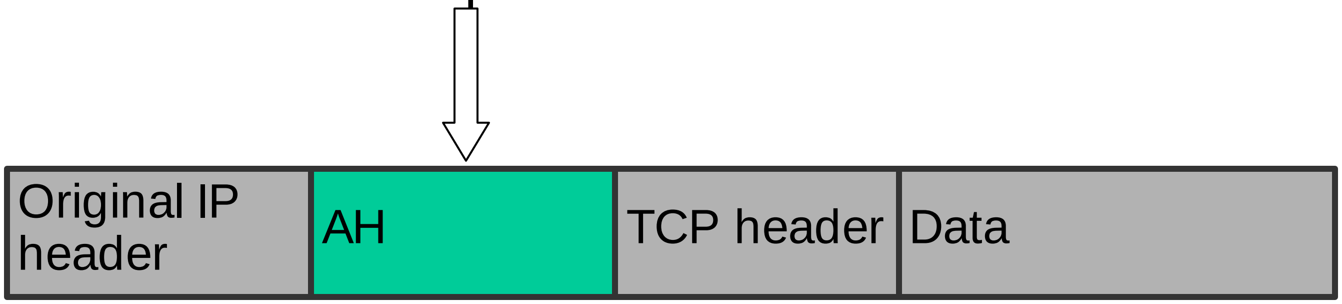
# Receiving IPsec Packets

- When Bob receives a message:
- Lookup the SA based on the *destination* address and SPI (in a multicast message the address is not Bob's own)
  - If the packet is unsecured (no IPsec) search through SPD for match—if no matching entry or if policy is PROTECT or DISCARD, the packet is discarded
- Find algorithm, key, sequence number, etc.
- After decrypting message, deliver packet to the next higher layer (such as TCP)

# Encapsulation Formats

- AH
  - Authentication Header
  - Provides integrity
- ESP
  - Encapsulating Security Payload
  - Provides integrity and/or privacy

AH in transport mode



# AH: Authentication Header

- RFC 4302
- Sender authentication
- Integrity for packet contents and IP header
- Sender and receiver must share a secret key
  - This key is used in HMAC computation (message authentication code computed with a hash)
  - The key is set up by IKE key establishment protocol and recorded in the Security Association (SA)

**AHv2, RFC 4302**

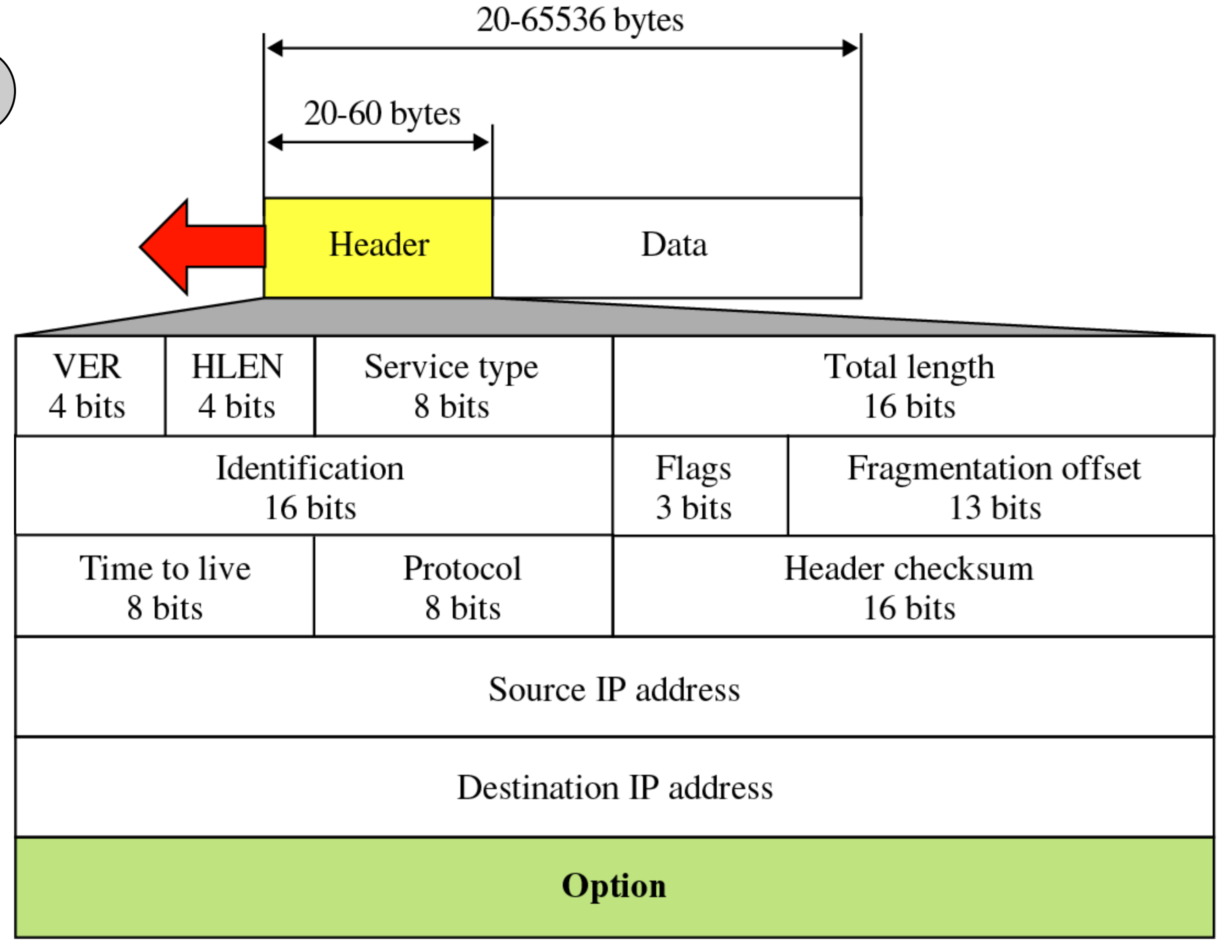
Let authentication header implement IP integrity by holding a hash of a shared secret and the content of an IP packet

# AH and IP Header

**Mutable fields may change:**
Service type, Fragm. Offset, TTL, Header checksum

**Predictable fields may change in a predictable way:**
Dst address (source routing)

**Immutable fields will not change:**
The rest....

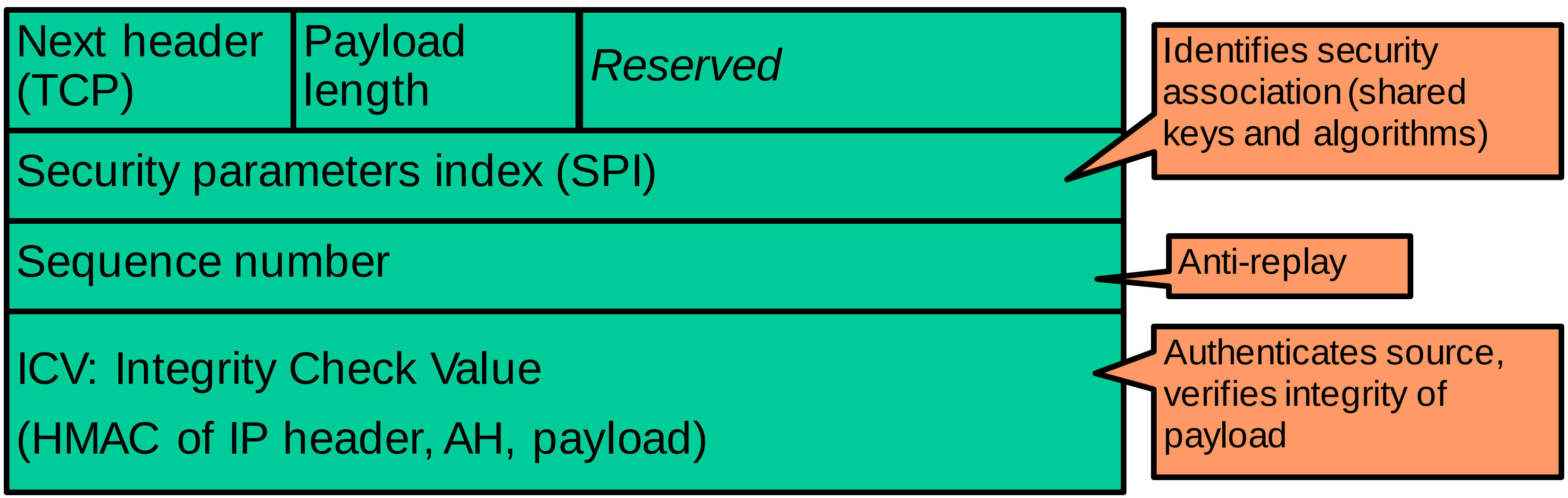





Mutable fields can’t be included in the AH’s end-to-end integrity check

# Authentication Header Format

- Provides integrity and origin authentication
- Authenticates portions of the IP header
- Anti-replay service (to counter denial of service)
- No confidentiality

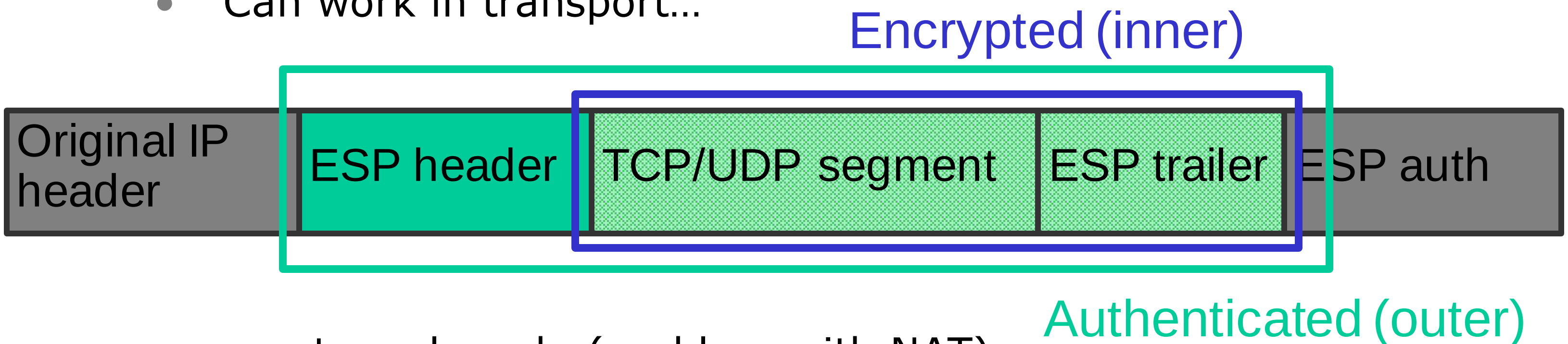


# ESP: Encapsulating Security Payload

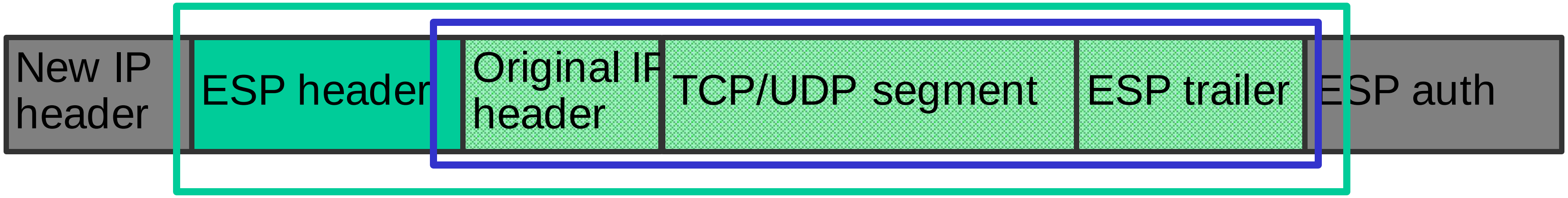
- RFC 4303
- Adds new header and trailer fields to packet
- Transport mode
  - Confidentiality of packet between two hosts
  - Complete hole through firewalls (for IPsec from a particular IP address)
  - Used sparingly
- Tunnel mode
  - Confidentiality of packet between two gateways or a host and a gateway
  - Implements VPN tunnels
  - FW filtering can be done on packets before they enter tunnel

# ESP Security Guarantees

- Confidentiality and integrity for packet payload
  - Symmetric cipher negotiated as part of security assoc
- Optionally provides authentication (similar to AH)
- Can work in transport…



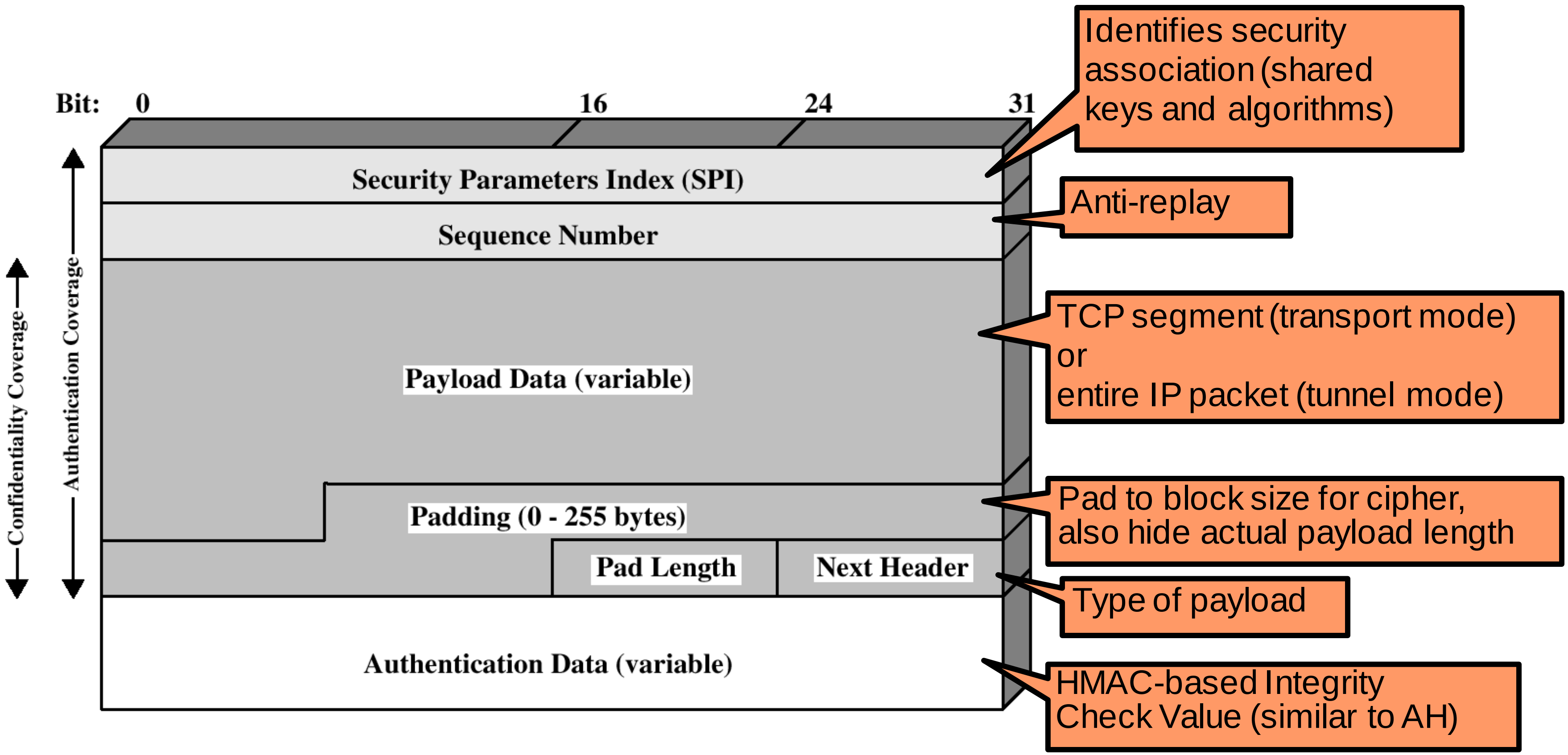

- …or tunnel mode (problem with NAT)



# Tunnel Mode and NAT

- Tunnel mode can be problematic together with NAT
- If we set up a tunnel between our host and a public gateway, it won't work:
  - Our private addresses will be in the original IP header
- It is OK to set up a tunnel between our host and a private intranet:
  - Private intranet addresses will be in the original IP header
  - New IP header will contain our home private address, which will be translated by the NAT

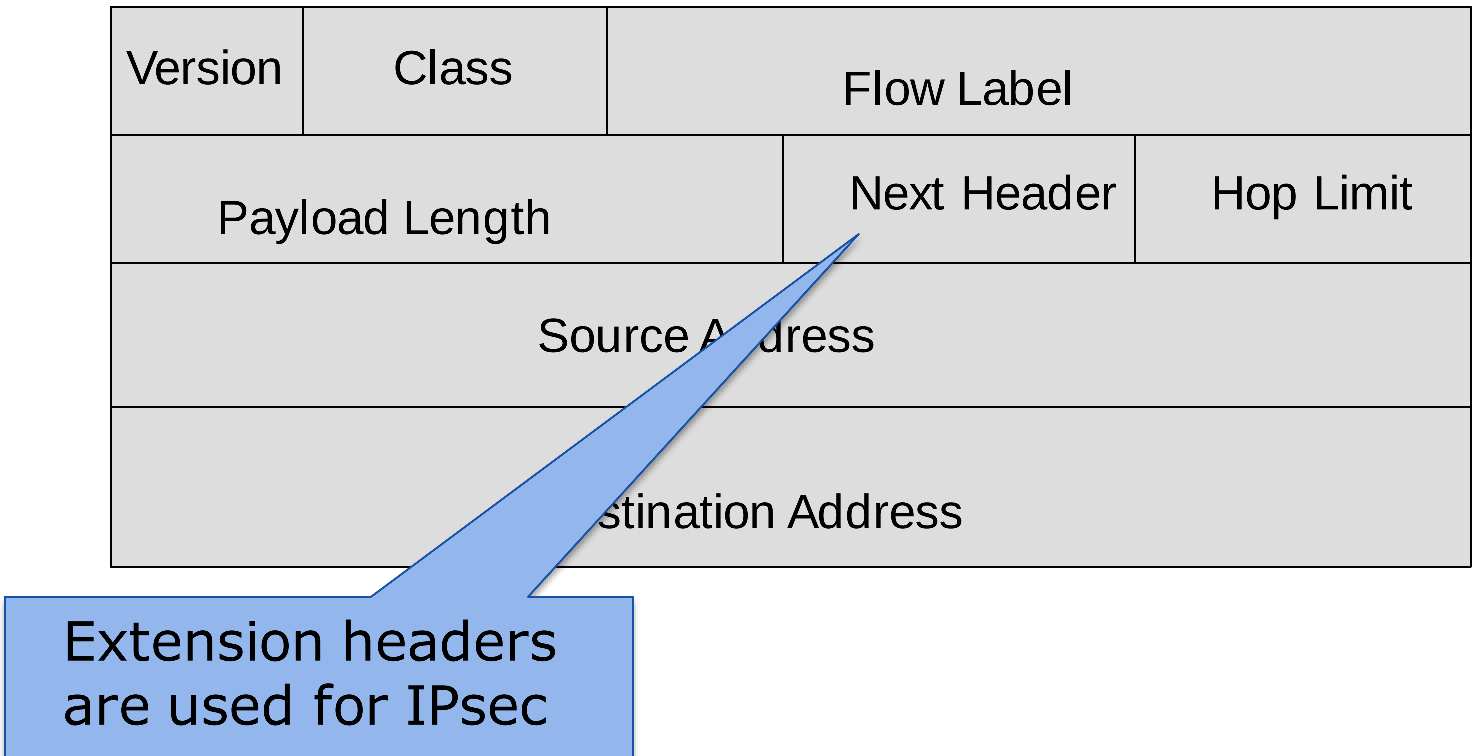
# ESP Packet



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
Identifies security association (shared keys and algorithms)
Anti-replay
TCP segment (transport mode) or entire IP packet (tunnel mode)
Pad to block size for cipher, also hide actual payload length
Type of payload
HMAC-based Integrity Check Value (similar to AH)

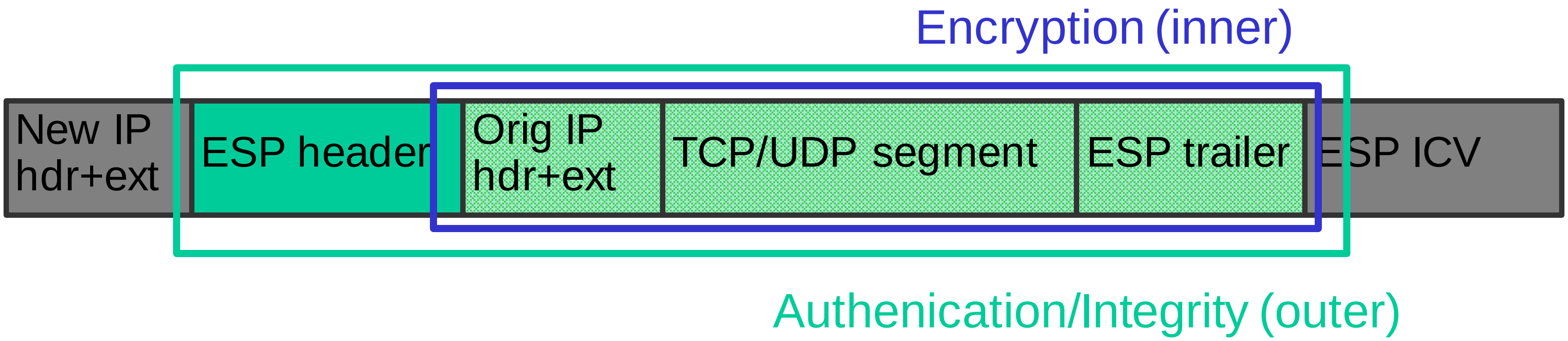
# IPsec and IPv6

- IPsec is a mandatory component for IPv6
- IPv6 header:



# IPsec Tunnel Mode in IPv6

- IPv6 IPsec is implemented using
  - Authentication extension header
  - ESP extension header

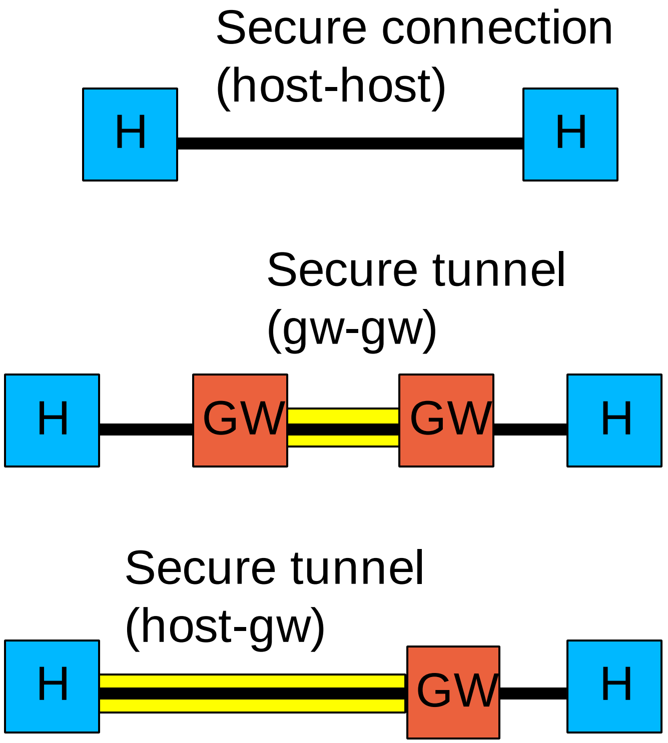


# Virtual Private Networks (VPN)

- ESP is often used to implement a VPN
  - Packets go from internal network to a gateway with TCP/IP headers for address in another network
  - Entire packet hidden by encryption
    - Including original headers so destination addresses are hidden
  - Receiving gateway decrypts packet and forwards original IP packet to receiving address in the network that it protects
- This is known as a VPN tunnel
  - Secure communication between parts of the same organization over public Internet
- The term IPsec VPN is sometimes used for secure VPNs in general
  - Even though they don’t use the IPsec protocols…

# Use Cases Summary

- Host-Host
  - Transport mode
  - (Or tunnel mode)
- Gateway-Gateway
  - Tunnel mode
- Host-Gateway
  - Tunnel mode



ROYAL INSTITUTE
OF TECHNOLOGY

# Part 2
# IPsec: IKE

## Internet key exchange

# Secure Key Establishment

- Goal: generate and agree on a session key using some public initial information
- What properties are needed?
  - Authentication (know identity of other party)
  - Secrecy (generated key not known to any others)
  - Forward secrecy (compromise of one session key does not compromise of keys in other sessions)
  - Prevent replay of old key material
  - Prevent denial of service
  - Protect identities from eavesdroppers

# IKE

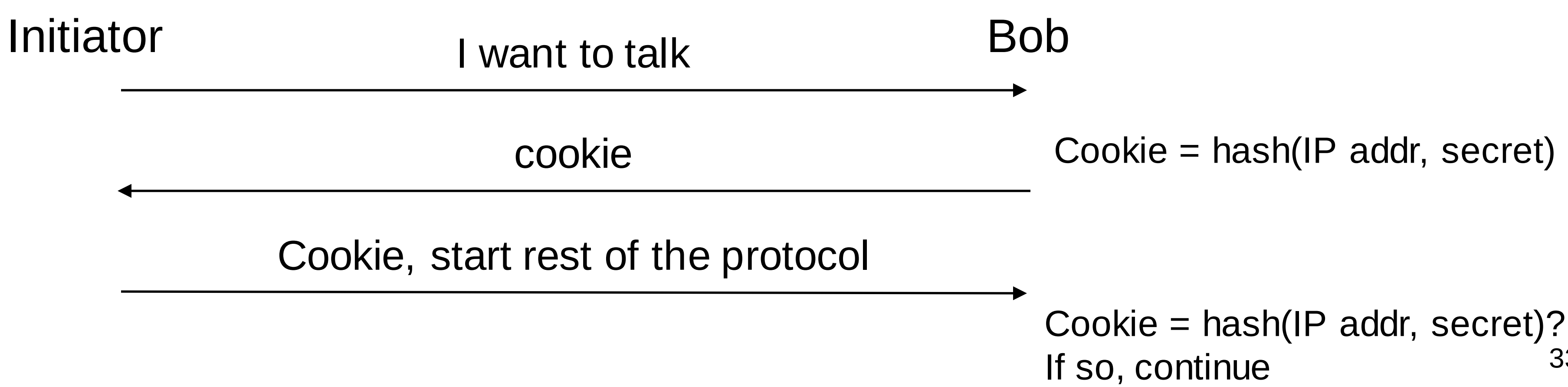
- Internet Key Exchange—setting up the SAs for IPsec (ESP and AH SA's)
- We assume that the two nodes have some long term key
  - Pre-shared secret key
  - Public encryption key
  - Public signature key
- Use IKE protocol to do mutual authentication and to create a session key
  - Use Diffie-Hellman to derive shared symmetric key
- IKE does not define exactly which ciphers to use, but a mechanism in which the nodes will negotiate this

# Diffie-Hellman

- Secret keys are created only when needed
  - No need to store secret keys for a long period of time, exposing them to increased vulnerablility
- Exchange requires no preexisting infrastructure
  - other than an agreement on the global parameters
    - A large prime number, p
    - A primitive root of p, g
    - Each partie has its own secret: a and b respectively
    - Secret shared key is $g^{ab}$ mod p
- For IKE to use Diffie-Hellman we need to add
  - Cookies for protection against denial-of-service attacks
  - Nonces to ensure against replay attacks
    - A number any given user of a protocol uses only once (large random number or sequence number, for instance)

# Cookies for Key Management in IPsec

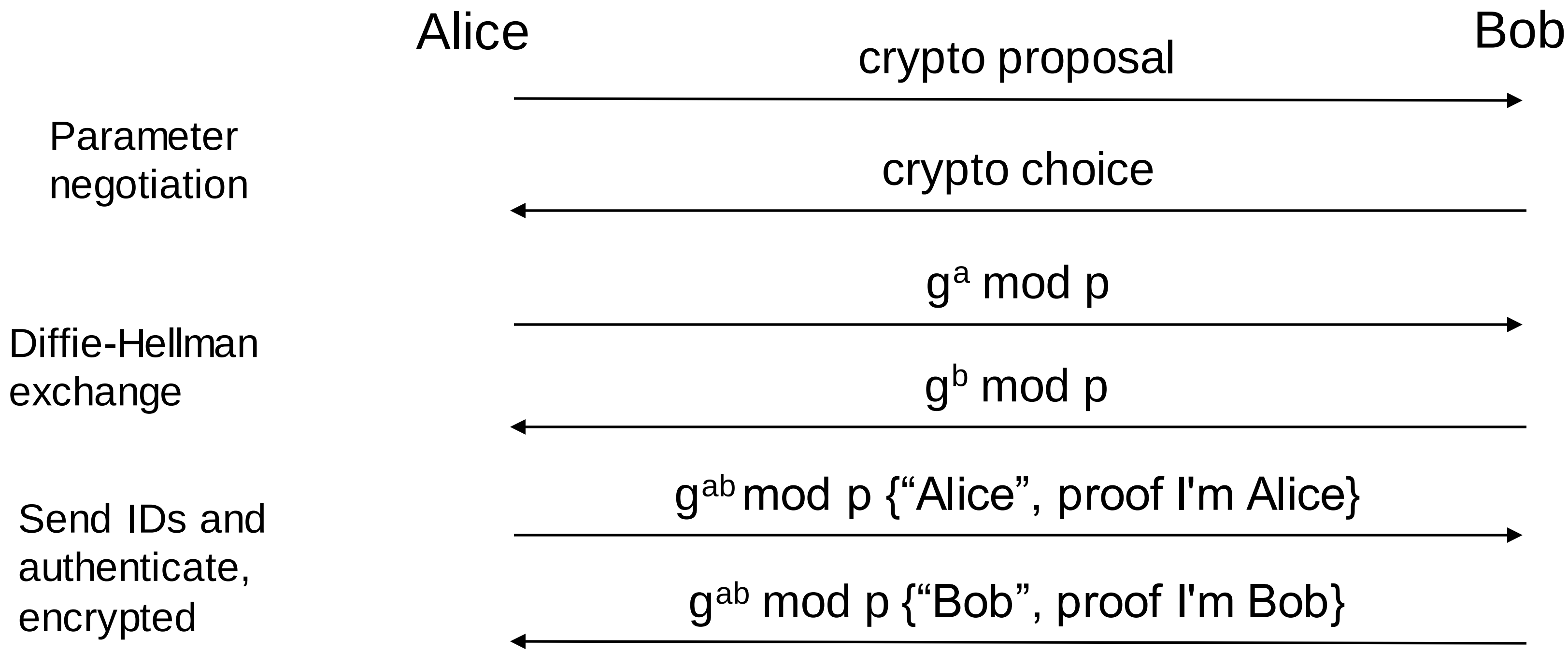
- Protect against denial-of-service/clogging
  - Impostor launches the attack packets with forged IP src addr
- Solution:
  - When Bob receives connection initiation from IP addr S
    - Send unpredictable number (cookie) to S—should be stateless!
    - Do nothing until same cookie is received from S
  - Assures that initiator can receive packets sent to S



# IKE Phases

- Phase 1
  - do mutual authentication and establish IKE session keys
  - Sets up the “main” SA (or IKE SA)
- Phase 2
  - Set up one or more IPsec SAs (child SAs) between the nodes using the keys derived in phase 1
- Why two phases?
  - Mutual authentication is expensive
  - If multiple SAs are needed or if SA parameters need to be changed, this can be done without repeating mutual authentication

# IKE Phase 1—Main Mode




- Proof of identity different for different key types
  - Pre-shared secret, private encryption or signature key,...
- Proof is a hash of
  - key, Diffie-Hellman values, nonces, crypto choices, cookies

# IKE Phase 1—Main Mode cont,d

- More details: cookies and nonces

Alice | Bob

crypto proposal + initiator cookie

crypto choice + initiator cookie, responder cookie

$g^a$ mod p + cookie pair + $\text{nonce}_A$

$g^b$ mod p + cookie pair + $\text{nonce}_B$

$g^{ab}$ mod p {“Alice”, proof I'm Alice}

$g^{ab}$ mod p {“Bob”, proof I'm Bob}

Recommended method for creating the cookie:

– Fast hash (e.g., MD5) over
  - IP src/dst addr, UDP src/dst port, locally generated secret value

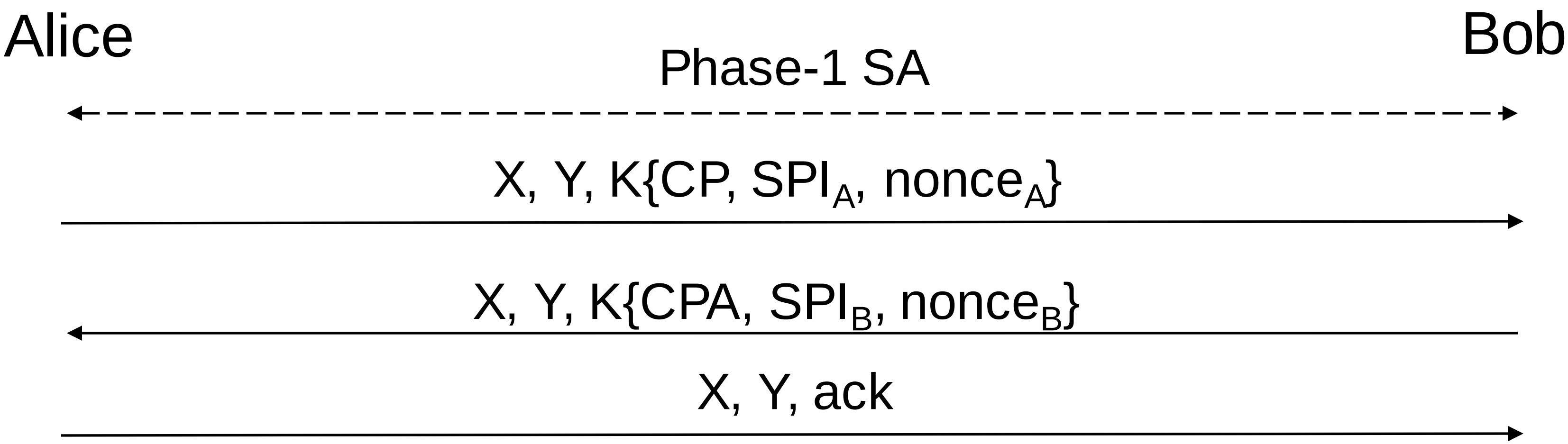
# IKE Phase 1—Session keys

$g^{ab} \bmod p$ {"Alice", proof I'm Alice}

Means encrypted

- Previous pictures show the general idea
- What are the actual *session keys*?
  - Integrity key and encryption key
- To calculate various keys:
  - IKE first calculates a quantity known as SKEYID
    - Prf(nonces, cookeis, Diffie-Hellman values)
  - IKE then calculates secret bits called SKEYID_d
    - Prf(SKEYID, and some other values)
  - Use Prf again to create SKEYID_a and SKEYID_e
    - a = authentication and e = encryption

# IKE Phase 2, Setting up IPsec SAs




- X is a pair of cookies from phase 1
- Y is a 32-bit number
  - ID to distinguish between multiple phase 2 sessions
- The rest is encrypted using SKEYID_e and authenticated using SKEYID_a
  - This part is simplified—more info can be exchanged

# IKEv2

- IKE has a history
  - ISAKMP (RFC 2408): framework rather than protocol
  - OAKLEY (RFC 2412) and SKEME: protocols working within ISAKMP
  - IKE (RFC 2409)
- IKEv2 (RFC 5996)
  - One single document for the standard
  - Simpler message exchange
  - Increased robustness (avoiding deadlocks)
  - Supporting NAT traversal
  - Supporting mobility
  - Supporting SCTP

ROYAL INSTITUTE
OF TECHNOLOGY

# Thanks for listening