

IPsec

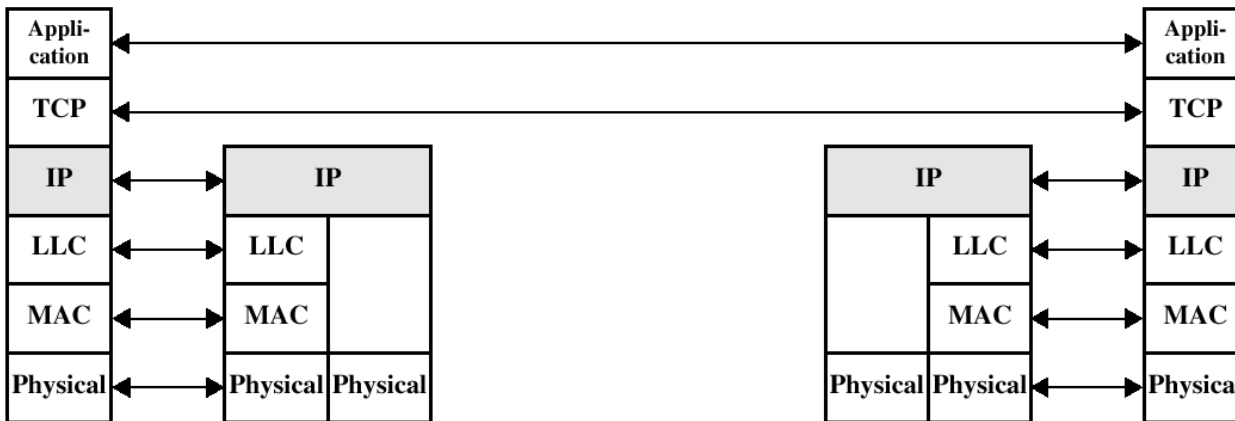
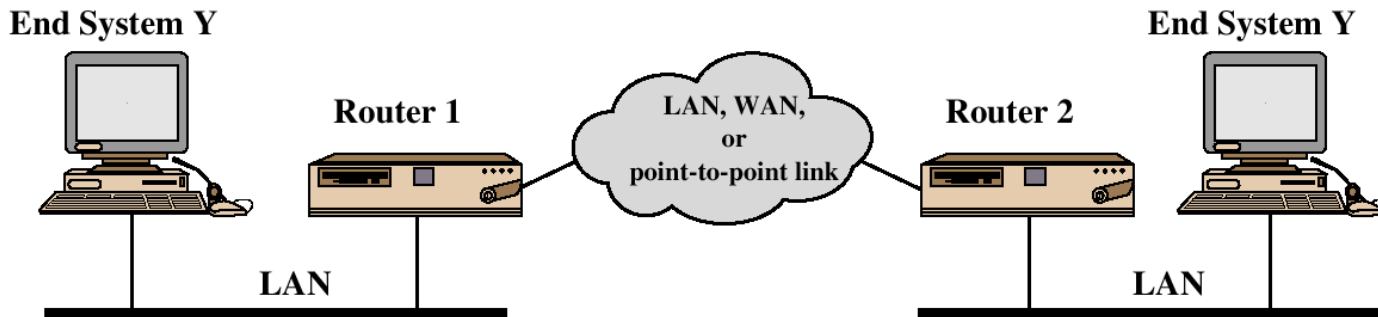


Vitaly Shmatikov

Reading Assignment

◆ Kaufman. Chapters 17 and 18.

TCP/IP



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 every protocol running on top of IPv4 and IPv6

IPsec: Network Layer Security

IPsec = AH + ESP + IPcomp + IKE

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

Compression

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 partial sequence integrity (prevent packet replay)
- ◆ Confidentiality (encapsulation) for packet contents
 - Also partial protection against traffic analysis
- ◆ Authentication and encapsulation can be used separately or together
- ◆ Either provided in transport or tunnel mode
- ◆ These services are **transparent** to applications above transport (TCP/UDP) layer

IPsec Modes

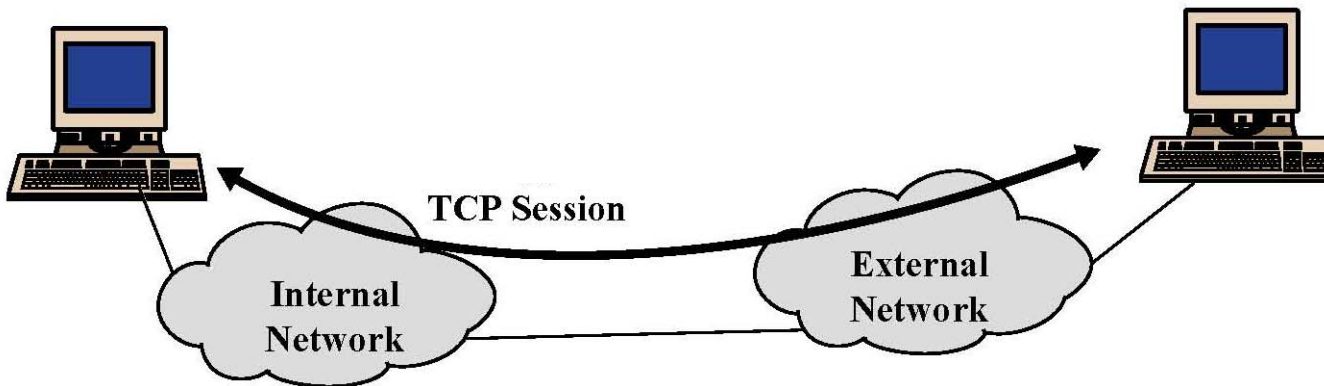
◆ Transport mode

- Used to deliver traffic from host to host or from host to gateway
- End-to-end across networks or within same network

◆ Tunnel mode

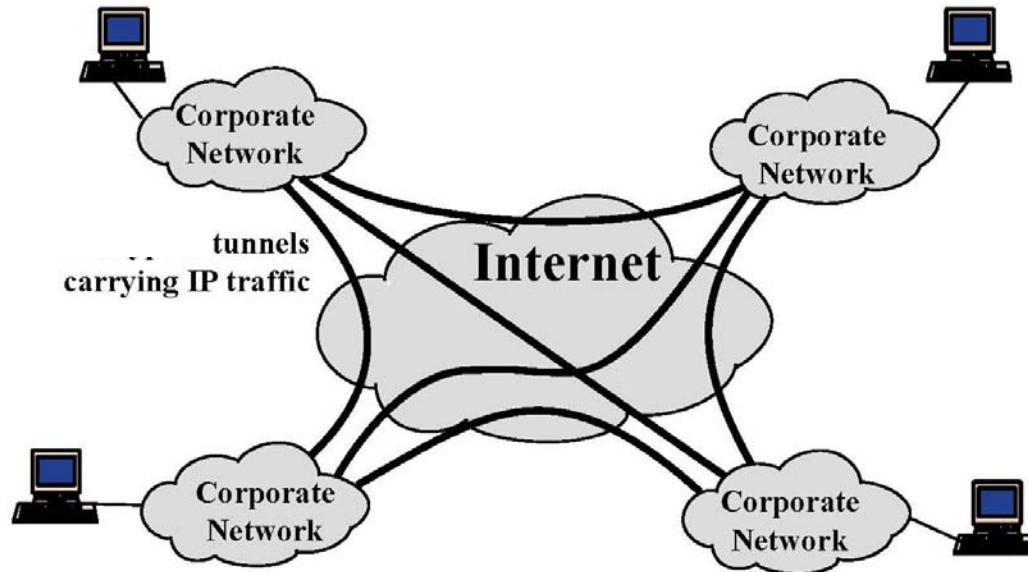
- Used to deliver traffic from gateway to gateway or from host to gateway
- Usually gateways are owned by the same organization
 - With an insecure network in the middle

IPsec in Transport Mode



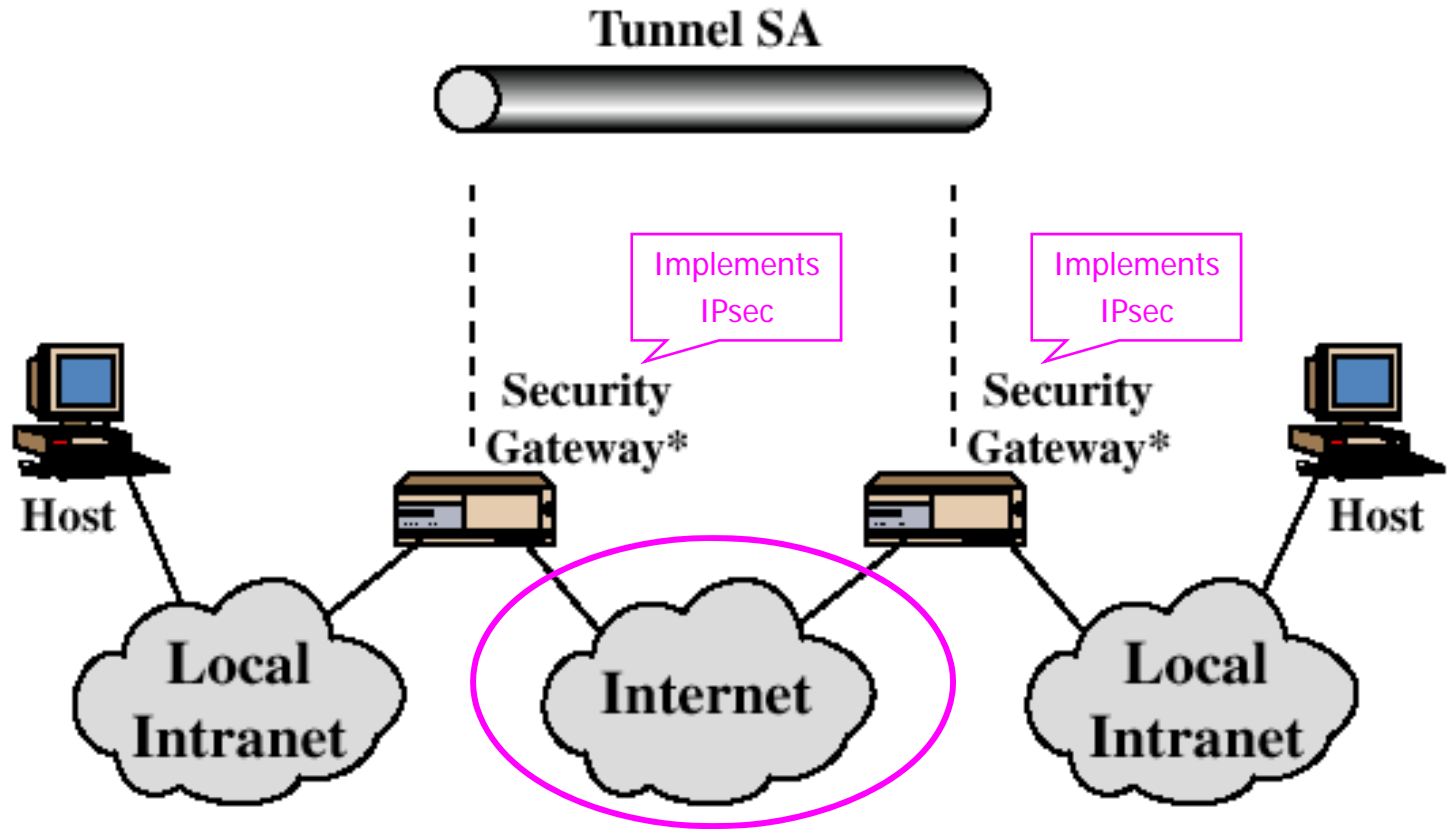
- ◆ End-to-end security between two hosts
 - Typically, client to gateway (e.g., PC to remote host)
- ◆ Requires IPsec support at each host

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

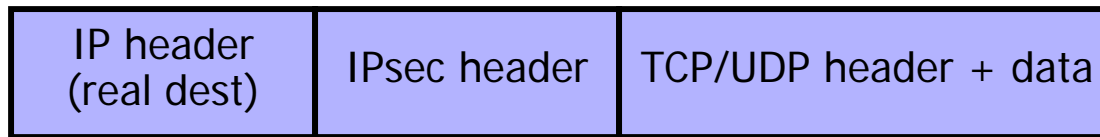
Tunnel Mode Illustration



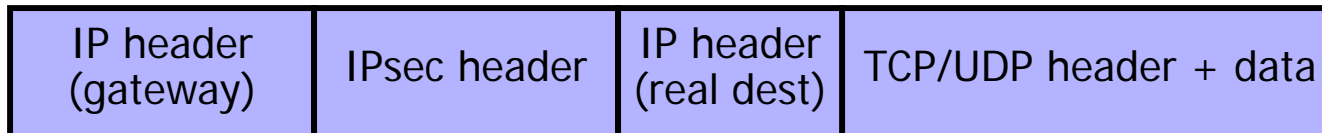
IPsec protects communication on the insecure part of the network

Transport Mode vs. Tunnel Mode

- ◆ **Transport mode** secures packet payload and leaves IP header unchanged



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



Security Association (SA)

- ◆ One-way sender-recipient relationship
 - Two SAs required for a two-way conversation
- ◆ SA determines how packets are processed
 - Cryptographic algorithms, keys, IVs, lifetimes, sequence numbers, mode (transport or tunnel) – read textbook!
- ◆ SA is uniquely identified by SPI (Security Parameters Index)...
 - Each IPsec implementation keeps a database of SAs
 - SPI is sent with packet, tells recipient which SA to use
- ◆ ...destination IP + protocol identifier (AH or ESP)

AH: Authentication Header

- ◆ Sender authentication
- ◆ Integrity for packet contents and IP header
- ◆ Sender and receiver must share a secret key
 - This key is used in HMAC computation
 - The key is set up by IKE key establishment protocol and recorded in the Security Association (SA)
 - SA also records protocol being used (AH) and mode (transport or tunnel) plus hashing algorithm used
 - MD5 or SHA-1 supported as hashing algorithms

IP Headers

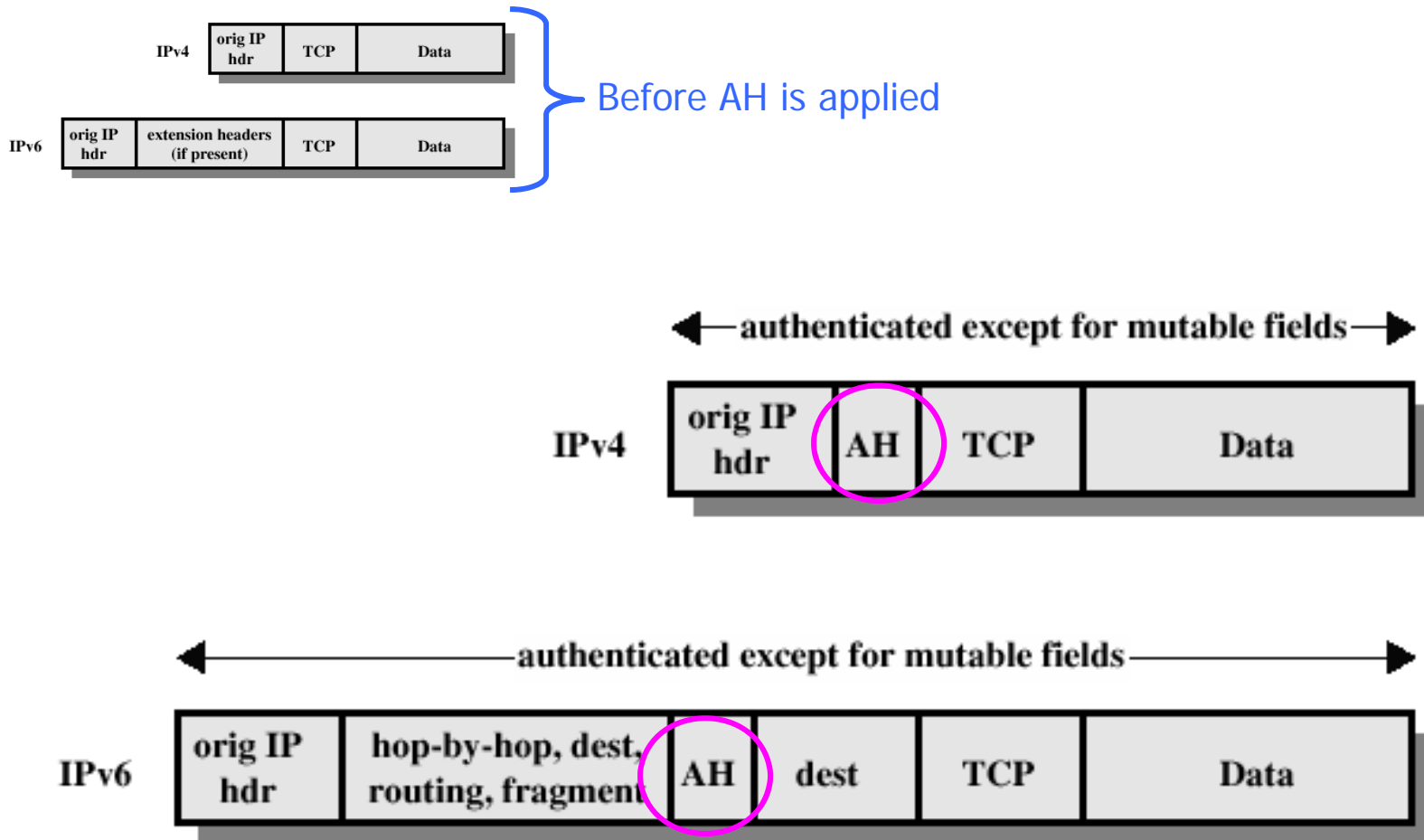


| | | | | | |
|---------|---------------|-----|---------------|-----------|-------|
| Version | Header Length | TOS | Packet length | Packet Id | Flags |
|---------|---------------|-----|---------------|-----------|-------|

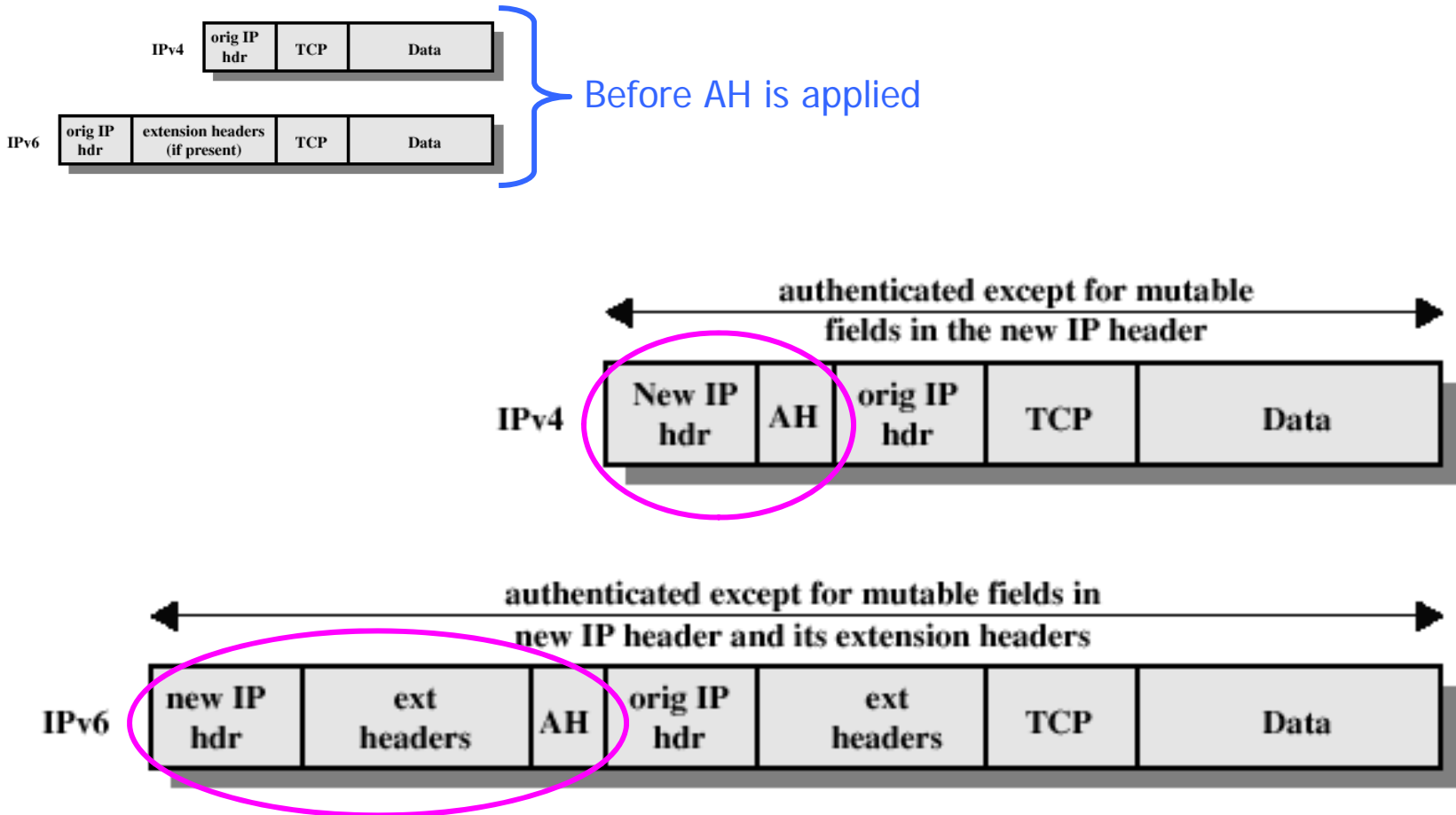
| | | | | | | |
|-----------------|-----|-----------------|----------|-------------------|------------------------|---------|
| Mutable | | Immutable | | Predictable | | |
| Fragment offset | TTL | Protocol number | Checksum | Source IP address | Destination IP address | Options |

AH sets mutable fields to zero and predictable fields to final value and then uses this header plus packet contents as input to HMAC

AH in Transport Mode

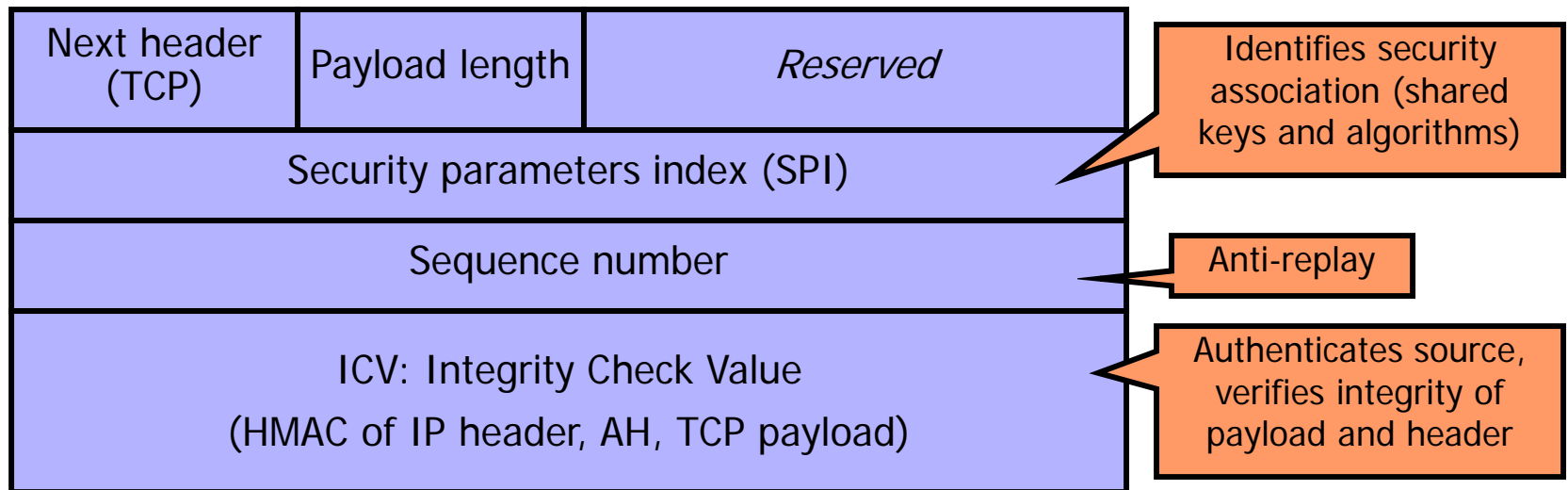


AH in Tunnel Mode



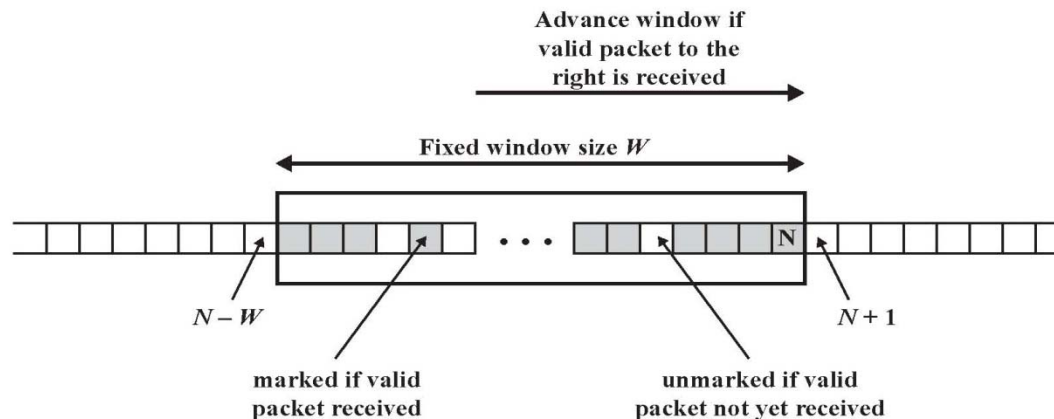
Authentication Header Format

- ◆ Authenticates origin of the packet
- ◆ Integrity for payload and parts of IP header
- ◆ Anti-replay service (to counter denial of service)
- ◆ No confidentiality



Prevention of Replay Attacks

- ◆ When SA is established, sender initializes 32-bit counter to 0, increments by 1 for each packet
 - If wraps around $2^{32}-1$, new SA must be established
- ◆ Recipient maintains a sliding 64-bit window
 - If a packet with high sequence number is received, do not advance window until packet is authenticated

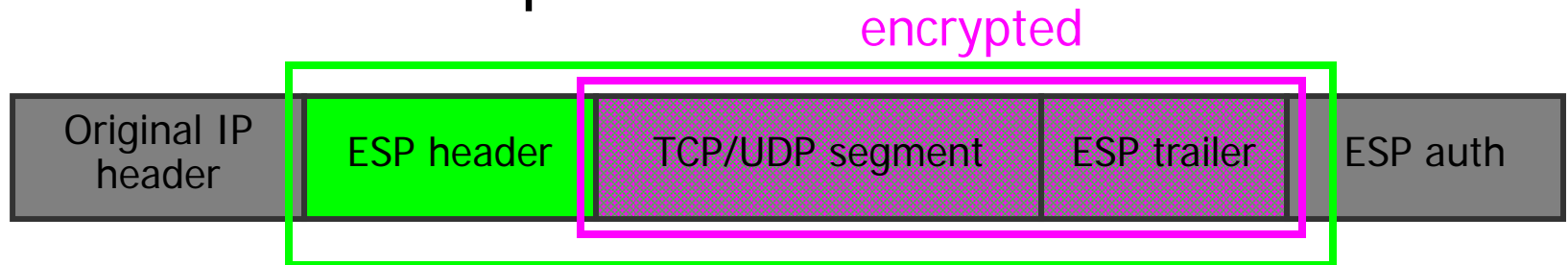


ESP: Encapsulating Security Payload

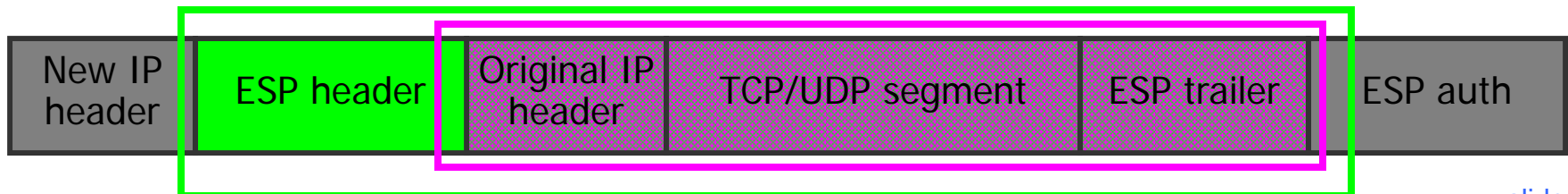
- ◆ Adds new header and trailer fields to packet
- ◆ Transport mode
 - Confidentiality of packet between two hosts
 - Complete hole through firewalls
 - Used sparingly
- ◆ Tunnel mode
 - Confidentiality of packet between two gateways or a host and a gateway
 - Implements VPN tunnels

ESP Security Guarantees

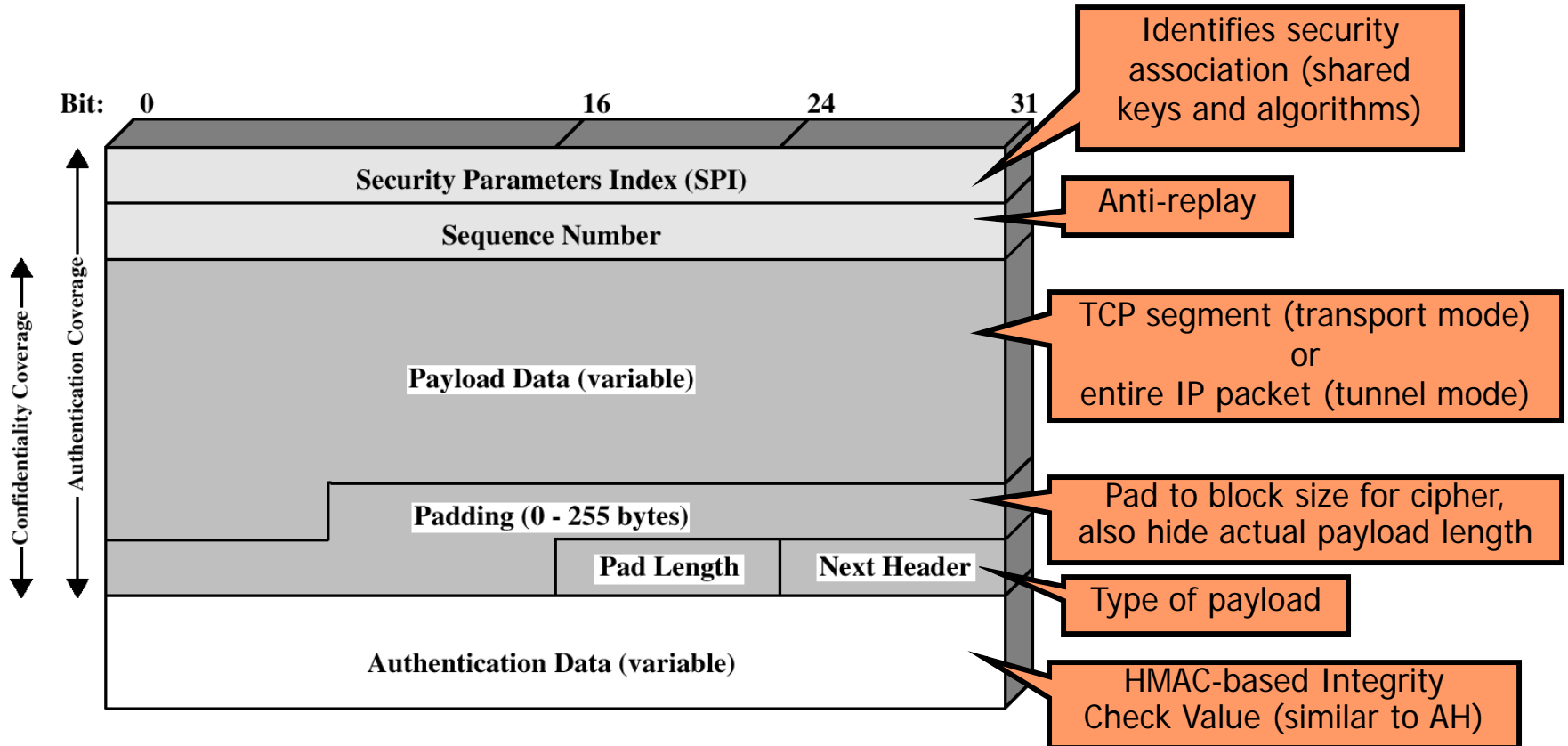
- ◆ **Confidentiality** and integrity for packet payload
 - Symmetric cipher negotiated as part of security assoc
- ◆ Optionally provides **authentication** (similar to AH)
- ◆ Can work in transport...



- ◆ ...or tunnel mode



ESP Packet

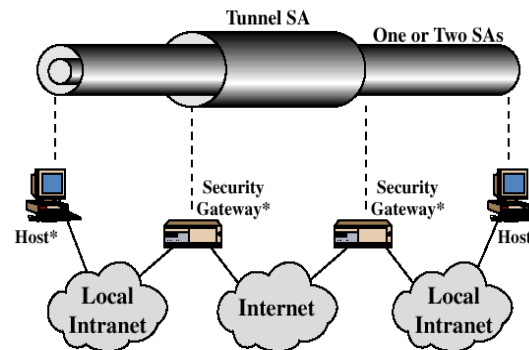


Virtual Private Networks (VPN)

- ◆ 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 untrusted Internet

Multiple SAs on a Single Packet

- ◆ AH and ESP are often combined
- ◆ End-to-end AH in transport mode
 - Authenticate packet sources
- ◆ Gateway-to-gateway ESP in tunnel mode
 - Hide packet contents and addresses on the insecure part of the network
- ◆ Significant cryptographic overhead
 - Even with AH



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 keys in other sessions)
 - Prevent replay of old key material
 - Prevent denial of service
 - Protect identities from eavesdroppers
 - Agreement (both parties agree on the value of key)

Key Management in IPsec

◆ Manual key management

- Keys and parameters of crypto algorithms exchanged offline (e.g., by phone), security associations established by hand

◆ Pre-shared symmetric keys

- New session key derived for each session by hashing pre-shared key with session-specific nonces
- Standard symmetric-key authentication and encryption

◆ Online key establishment

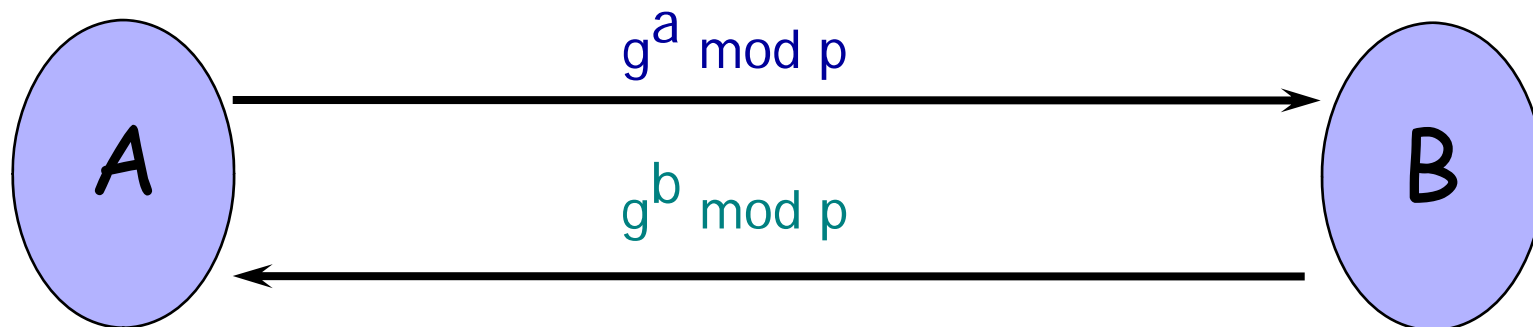
- Internet Key Exchange (IKE) protocol
- Use Diffie-Hellman to derive shared symmetric key

Diffie-Hellman Key Exchange

◆ Assume finite group $G = \langle S, \bullet \rangle$

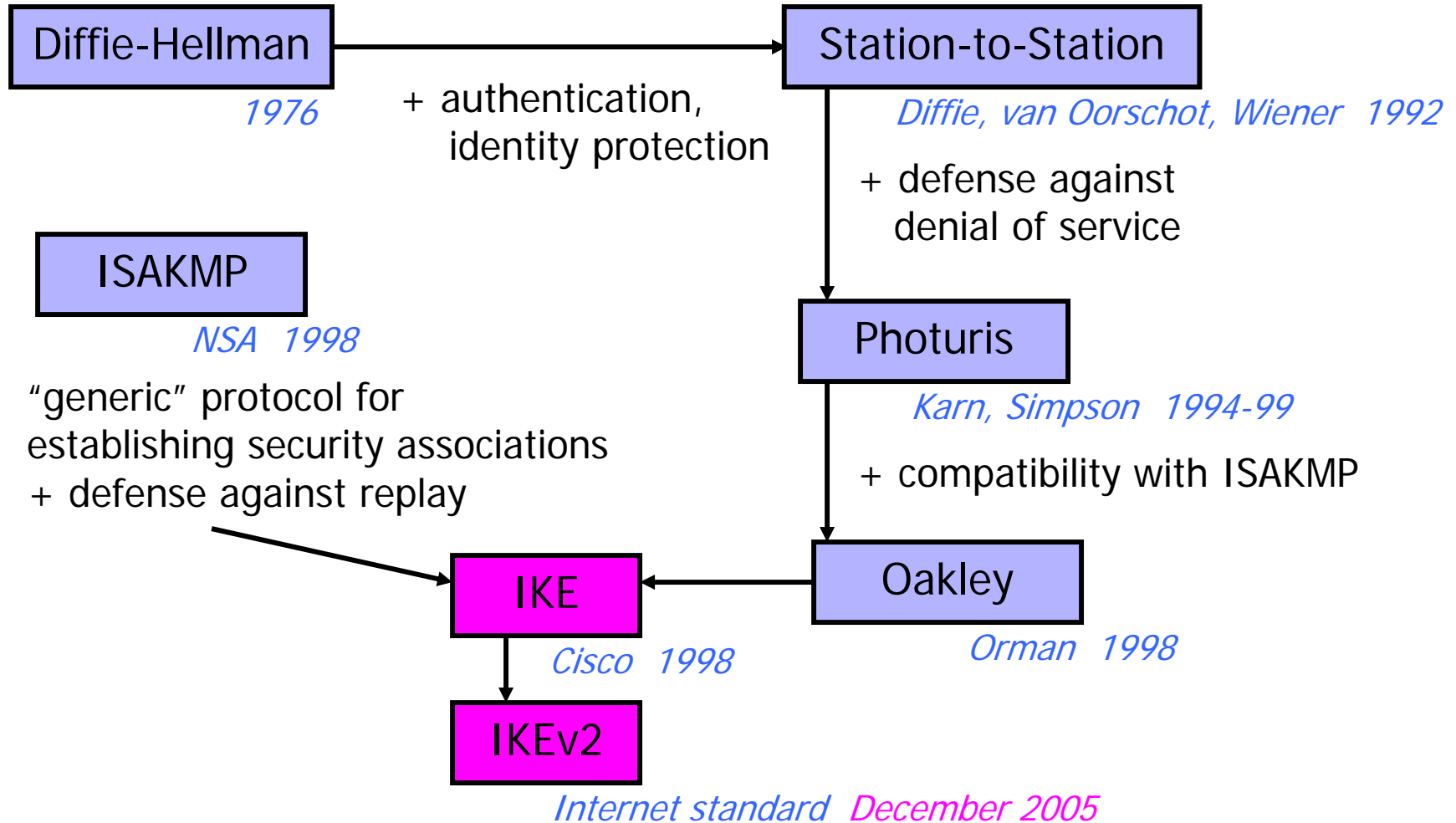
- Choose generator g so every $x \in S$ is $x = g^i$ for some i
- Example: squares modulo prime p (even i)

◆ Protocol



Alice, Bob share $g^{ab} \bmod p$ not known to anyone else

IKE Genealogy



Design Objectives

◆ Shared secret

- Create and agree on a secret which is known only to protocol participants

◆ Authentication

- Participants need to verify each other's identity

◆ Identity protection

- Eavesdropper should not be able to infer participants' identities by observing protocol execution

◆ Protection against denial of service

- Malicious participant should not be able to exploit the protocol to cause the other party to waste resources

Ingredient 1: Diffie-Hellman

A \rightarrow B: g^a

B \rightarrow A: g^b

- Shared secret is g^{ab} , compute key as $k = \text{hash}(\text{rand}, g^{ab})$
 - Diffie-Hellman guarantees perfect forward secrecy
- Authentication
- Identity protection
- DoS protection

Ingredient 2: Challenge-Response

A → B: m, A

B → A: n, sig_B(m, n, A)

A → B: sig_A(m, n, B)

- Shared secret
- Authentication
 - A receives his own number m signed by B's private key and deduces that B is on the other end; similar for B
- Identity protection
- DoS protection

DH + Challenge-Response

ISO 9798-3 protocol:

A \rightarrow B: g^a, A

B \rightarrow A: $g^b, \text{sig}_B(g^a, g^b, A)$

A \rightarrow B: $\text{sig}_A(g^a, g^b, B)$

| |
|------------|
| $m := g^a$ |
|------------|

| |
|------------|
| $n := g^b$ |
|------------|

- Shared secret: g^{ab}
- Authentication
- Identity protection
- DoS protection

Ingredient 3: Encryption

Encrypt signatures to protect identities:

A \rightarrow B: g^a, A

B \rightarrow A: $g^b, \text{Enc}_{\text{PK}(A)}(\text{sig}_B(g^a, g^b, A))$

A \rightarrow B: $\text{Enc}_{\text{PK}(B)}(\text{sig}_A(g^a, g^b, B))$

- Shared secret: g^{ab}
- Authentication
- Identity protection (for responder only!)
- DoS protection

Refresher: DoS Prevention

- ◆ Denial of service due to resource clogging
 - If responder opens a state for each connection attempt, attacker can initiate thousands of connections from bogus or forged IP addresses
- ◆ **Cookies** ensure that the responder is stateless until initiator produced at least two messages
 - Responder's state (IP addresses and ports) is stored in an unforgeable cookie and sent to initiator
 - After initiator responds, cookie is regenerated and compared with the cookie returned by the initiator
 - The cost is two extra messages in each execution

Refresher: Anti-DoS Cookie

◆ Typical protocol:

- Client sends request (message #1) to server
- Server sets up connection, responds with message #2
- Client may complete session or not (potential DoS)

◆ Cookie version:

- Client sends request to server
- Server sends hashed connection data back
 - Send message #2 later, after client confirms his address
- Client confirms by returning hashed data
- Need an extra step to send postponed message #2

Ingredient 4: Anti-DoS Cookie

“Almost-IKE” protocol:

A → B: g^a

B → A: g^b , $\text{hash}_{k_b}(g^b, g^a)$

A → B: g^a , g^b , $\text{hash}_{k_b}(g^b, g^a)$

$\text{Enc}_K(\text{sig}_A(g^a, g^b, B))$

B → A: g^b , $\text{Enc}_K(\text{sig}_B(g^a, g^b, A))$

Doesn't quite work: B must remember his DH exponent b for every connection

$k = \text{hash}(g^{ab})$

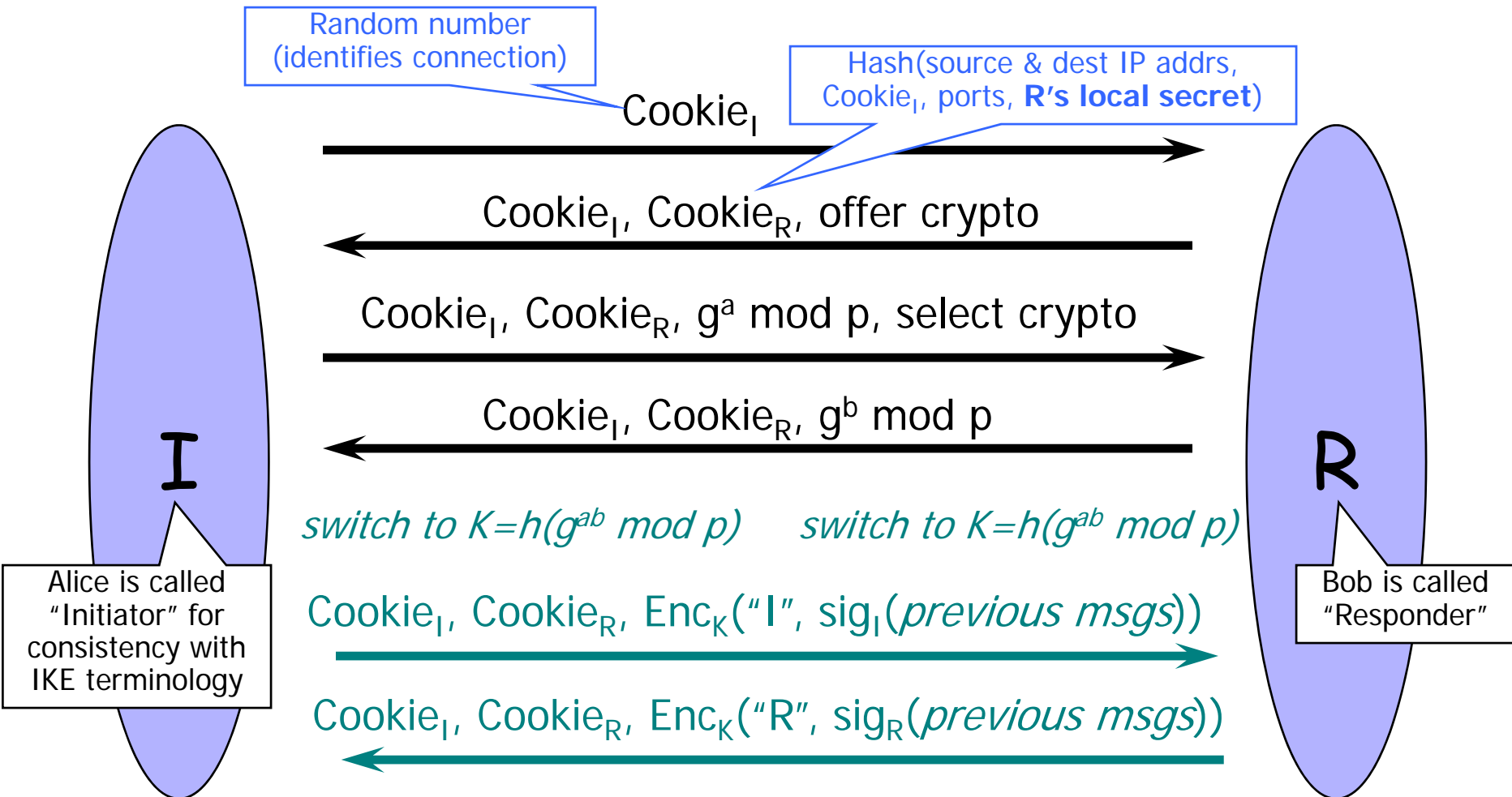
- Shared secret: g^{ab}
- Authentication
- Identity protection
- DoS protection?

Medium-Term Secrets and Nonces

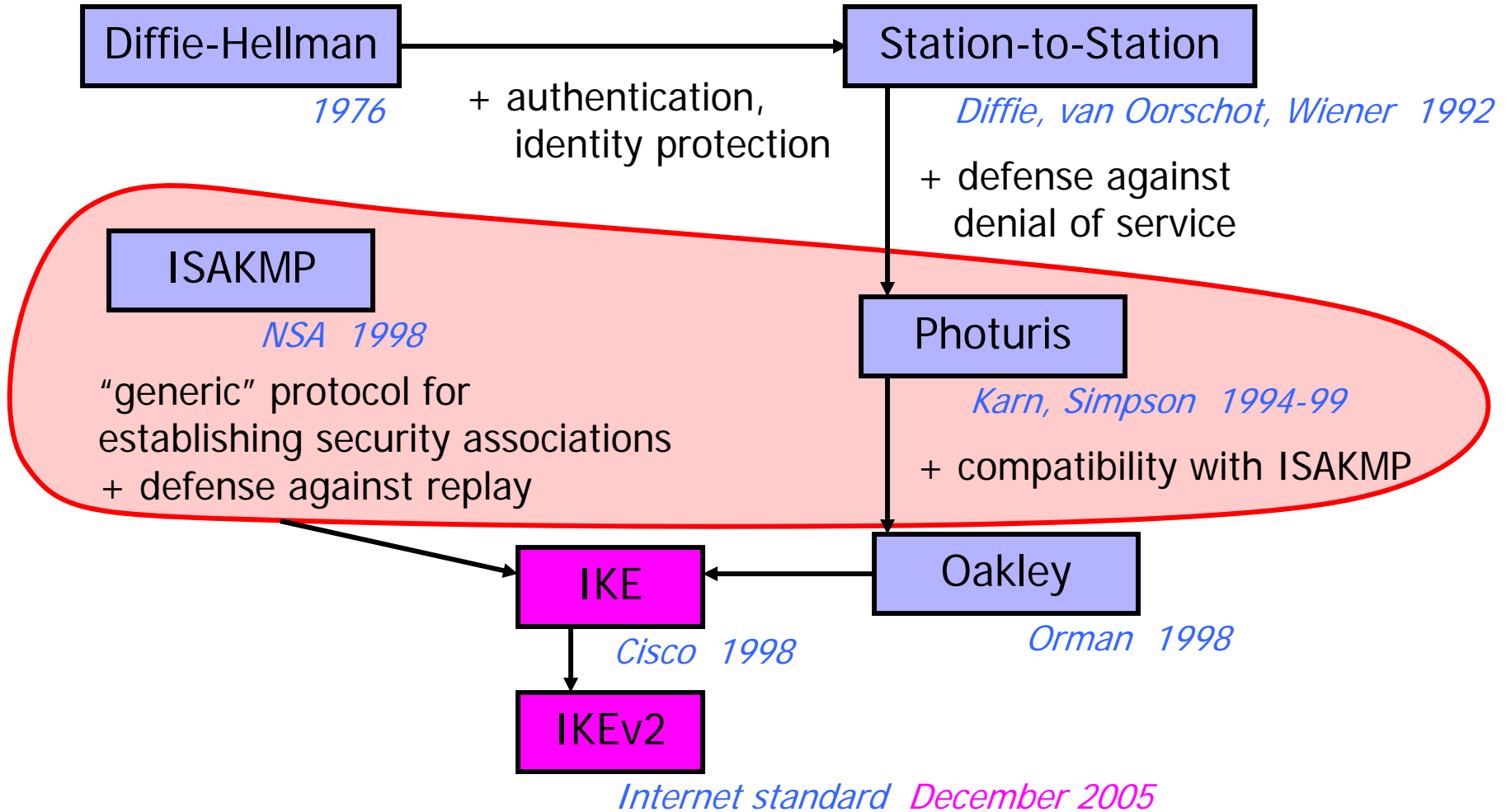
- ◆ Idea: use the same Diffie-Hellman value g^{ab} for every session, update every 10 minutes or so
 - Helps against denial of service
- ◆ To make sure keys are different for each session, derive them from g^{ab} and session-specific nonces
 - Nonces guarantee freshness of keys for each session
 - Re-computing g^a , g^b , g^{ab} is costly, generating nonces (fresh random numbers) is cheap
- ◆ This is more efficient and helps with DoS, but no longer guarantees forward secrecy (why?)

(Simplified) Photuris

[Karn and Simpson]



IKE Genealogy Redux



Cookies in Photuris and ISAKMP

- ◆ Photuris cookies are derived from local secret, IP addresses and ports, counter, crypto schemes
 - Same (frequently updated) secret for all connections
- ◆ ISAKMP requires unique cookie for each connect
 - Add timestamp to each cookie to prevent replay attacks
 - Now responder needs to keep state (“cookie crumb”)
 - Vulnerable to denial of service (why?)
- ◆ **Inherent conflict:** to prevent replay, need to remember values that you’ve generated or seen before, but keeping state allows denial of service

IKE Overview

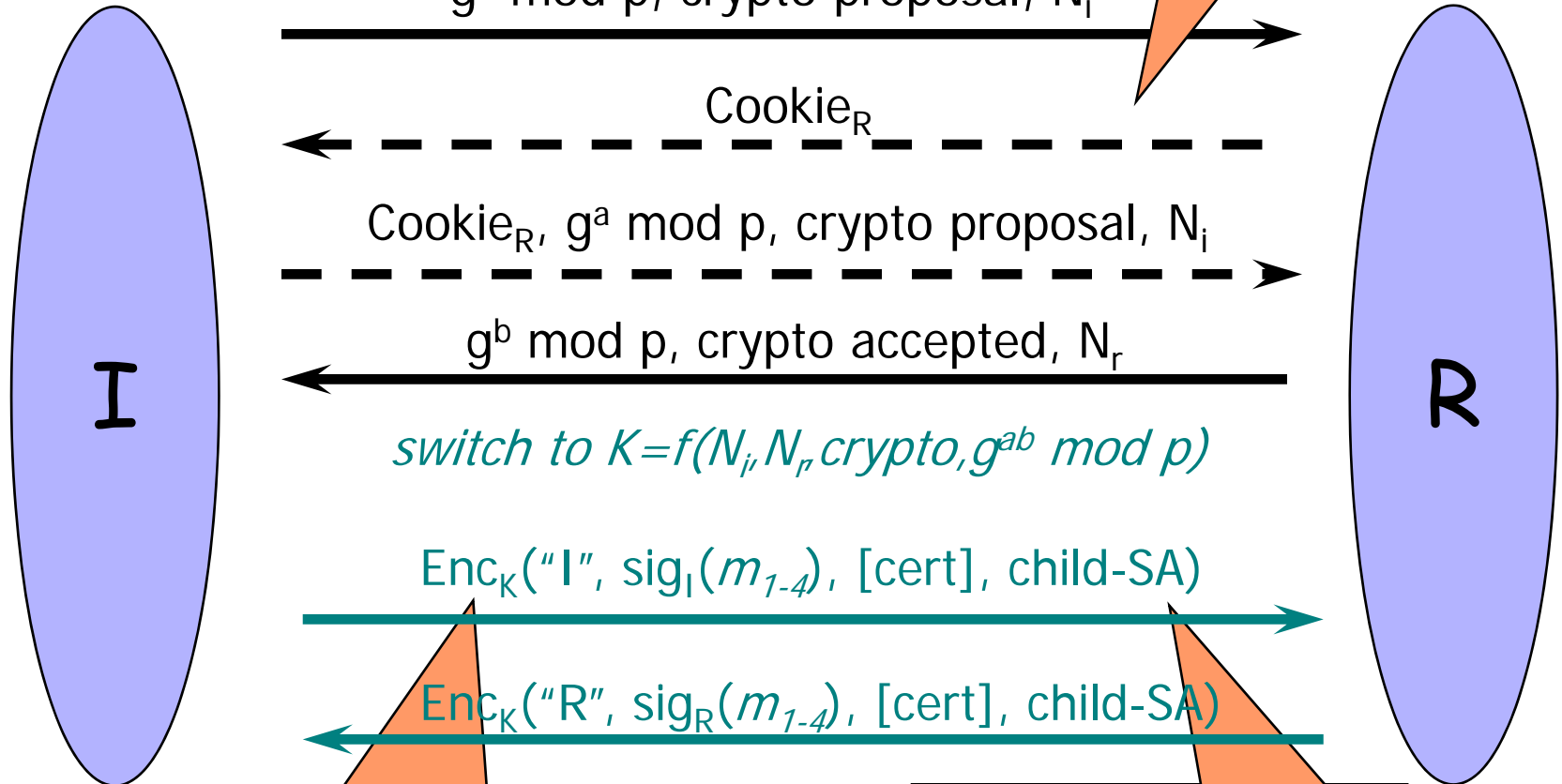
- ◆ Goal: create **security association** between 2 hosts
 - Shared encryption and authentication keys, agreement on crypto algorithms
- ◆ **Two phases**: 1st phase establishes security association (IKE-SA) for the 2nd phase
 - Always by authenticated Diffie-Hellman (expensive)
- ◆ 2nd phase uses IKE-SA to create actual security association (child-SA) to be used by AH and ESP
 - Use keys derived in the 1st phase to avoid DH exchange
 - Can be executed cheaply in “quick” mode
 - To create a fresh key, hash old DH value and new nonces

Why Two-Phase Design?

- ◆ Expensive 1st phase creates “main” SA
- ◆ Cheap 2nd phase allows to create multiple child SAs (based on “main” SA) between same 2 hosts
 - Example: one SA for AH, another SA for ESP
 - Different conversations may need different protection
 - Some traffic only needs integrity protection or short-key crypto
 - Too expensive to always use strongest available protection
 - Avoid multiplexing several conversations over same SA
 - For example, if encryption is used without integrity protection (bad idea!), it may be possible to splice the conversations
 - Different SAs for different classes of service

IKE: Phase One

Optional: refuse 1st message and demand return of stateless cookie

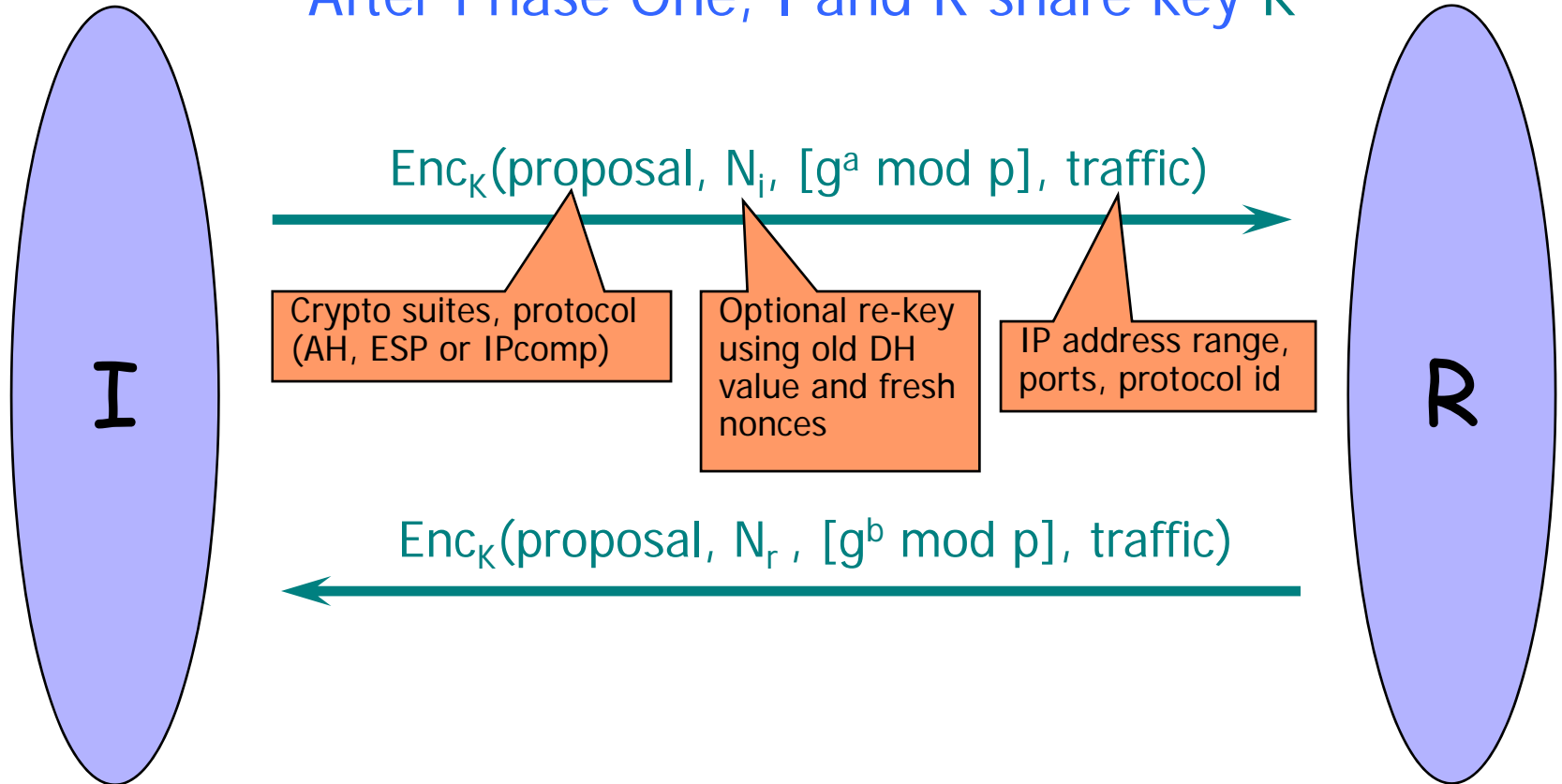


Initiator reveals identity first
Prevents "polling" attacks where attacker initiates IKE connections to find out who lives at an IP addr

Instead of running 2nd phase, "piggyback" establishment of child-SA on initial exchange

IKE: Phase Two (Create Child-SA)

After Phase One, I and R share key K



Can run this several times to create multiple SAs

Other Aspects of IKE

- ◆ Interaction with other network protocols
 - How to run IPsec through NAT (Network Address Translation) gateways?
- ◆ Error handling
 - Very important! Bleichenbacher attacked SSL by cryptanalyzing error messages from an SSL server
- ◆ Protocol management
 - Dead peer detection, rekeying, etc.
- ◆ Legacy authentication
 - What if one of the parties doesn't have a public key?

Current State of IPsec

- ◆ Best currently existing VPN standard
 - For example, used in Cisco PIX firewall, many remote access gateways
- ◆ IPsec has been out for a few years, but wide deployment has been hindered by complexity
 - ANX (Automotive Networking eXchange) uses IPsec to implement a private network for the Big 3 auto manufacturers and their suppliers