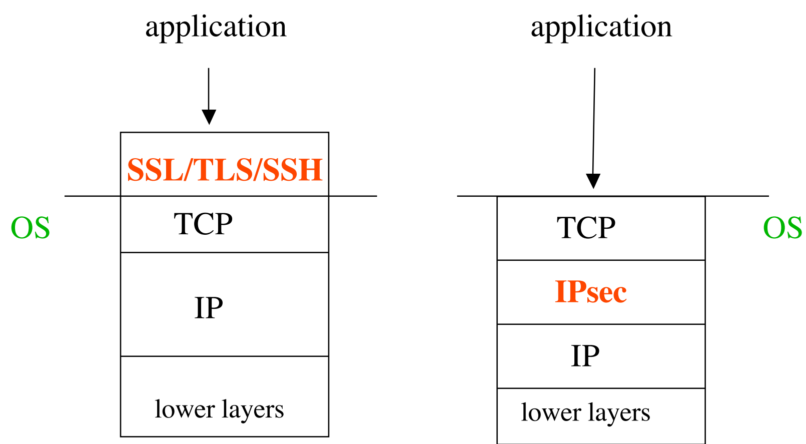
IPsec

IPsec as a Real-Time Protocol

- A real-time protocol is one where parties negotiate interactively to authentication each other and establish a session key
  - The conversation protected using the session key is called **security association**
- Examples: IPsec, SSL/TLS, SSH
  - Public key based

Security at Layer 4 vs. 3



Assumption: TCP/IP are in the OS

Pros and Cons

- Security at layer 4 (SSL/TLS/SSH)
  - \+ No need to change OS
  - – Applications have to be modified
  - – No way to tell TCP layer whether newly received data is bogus or real
    - Such as a sequence number attack
- Security at layer 3 (IPsec)
  - \+ Transparent to applications
  - – OS needs to modified
  - – Security is in terms of IP addresses
    - IPsec authentication cannot distinguish between users

IPsec User Model

- Alice and Bob sets up a secure channel
  - Called **Security Association**
- Then rely on IPsec to protect the channel

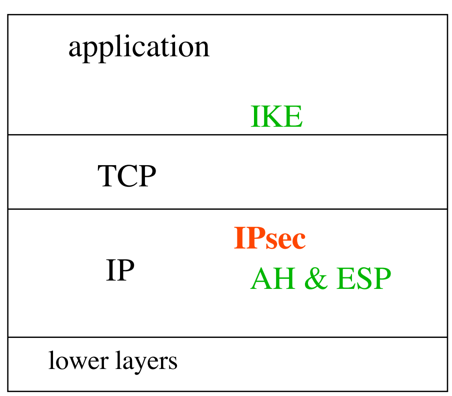
What does IPsec Accomplish?

- Encrypted traffic
- Connectionless Integrity
- Anti replay
- More secure authentication based on source IP address
- Enforced access control based on a policy database

- Similar to set up two firewalls between two ends

Main Pieces

- AH & ESP
  - IP header extensions for carrying cryptographically protected data
- IKE
  - A protocol for establishing security associations (SA) and establishing session keys
  - Not required for IPsec but recommended
    - IPsec also supports manual SAs/keying



IPsec Deployment

- Individual host: an end system can implement its own protection end-to-end or hop-by-hop
- Host community: a single security gateway (e.g. a firewall) can protect an entire domain of hosts
- Pairings: host-to-host, host-to-gateway, gateway-to-gateway
  - Or combined

Security Association

- An unidirectional cryptographically protected connection
  - Communication between Alice and Bob consists of two SAs, one for each direction
- Each end remembers:
  - Id of the other end
  - A cryptographic key
  - Sequence number currently being used
  - Cryptographic services being used
    - Integrity only, encryption only, or both
    - Which cryptographic algorithms

Copyright © 2003 Jun Li.
All rights reserved.

Security Association Database

- A security association database **(SAD)** is used to remember those info above for every **active** security association
  - Indexed by **security parameter index (SPI)**
- Thus an IPsec-capable node knows how to communicate with a given destination
  - A packet from Alice to Bob should tell Bob the SPI value that Bob can use to locate the Alice-Bob SA entry in his SAD

AH & ESP

- AH provides integrity protection
  - For payload and some fields in IP header
- ESP provides encryption and/or integrity protection
  - For payload
  - The encryption algorithm can be "null" or others

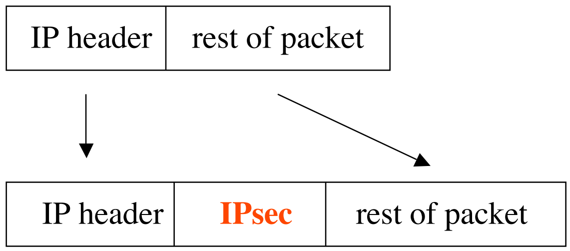
A Side Effect of IPsec on Firewall

- If a packet is protected using ESP, a firewall won't be able to inspect the payload of the packet
  - A firewall has even no idea whether the payload is encrypted or not
    - Recall the encryption algorithm could be "null"

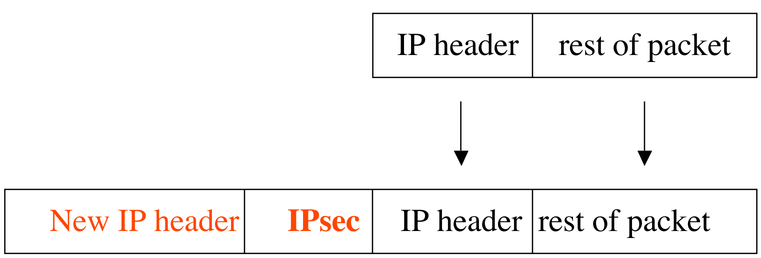
Two IPsec Modes

- Transport mode
- Tunnel mode

Transport Mode



Tunnel Mode



Mode Selection

- Transport mode is most logical when applying IPsec for end-to-end communication
- A tunnel mode is good for firewall-to-firewall, or end-to-firewall

An Example of Using Tunnel Mode

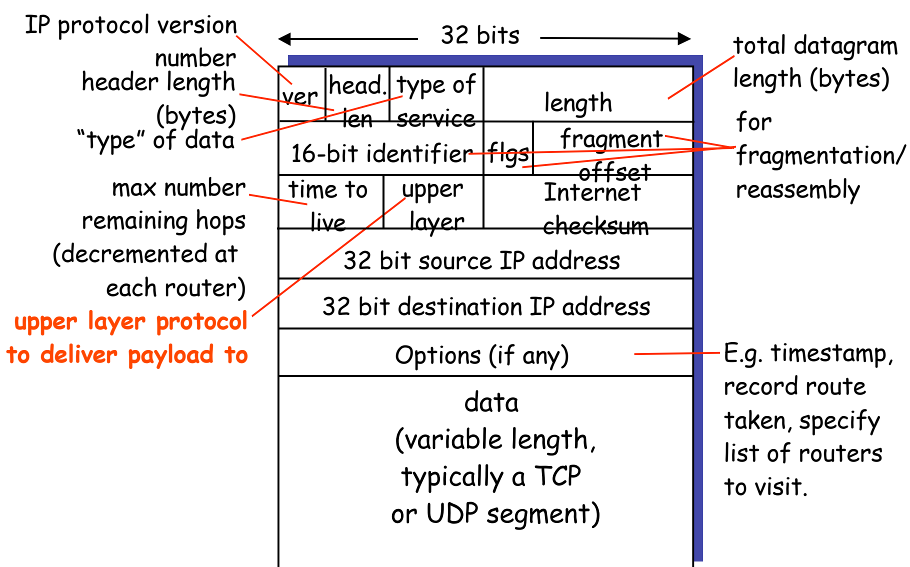
IP: src=F1, dst=F2	ESP	IP: src=A, dst=B	

Format of IPsec-Protected Packets

- A field in the IP header points to AH header or ESP header
  - "Protocol" field in IPv4
  - "Next header" field in IPv6

  - ESP = 50
  - AH = 51
  - (TCP = 6, UDP = 17)

IPv4 Datagram Format



AH - Authentication Header

# octets	
1	next header
1	payload length
2	unused
4	SPI (security parameter index)
4	sequence number
variable	authentication data

AH Fields

- Next header
  - Same as "protocol" field in IPv4
  - If TCP follows the AH header, this field is 6
- Payload length:
  - The size of the AH header (in 32-bit chunks)
- SPI
  - For the recipient to locate the SA entry in its SAD
- Sequence number:
  - For anti-replay purpose
- Authentication data
  - Cryptographic integrity check
  - Those immutable and mutable-but-predictable fields in an IP header are also protected

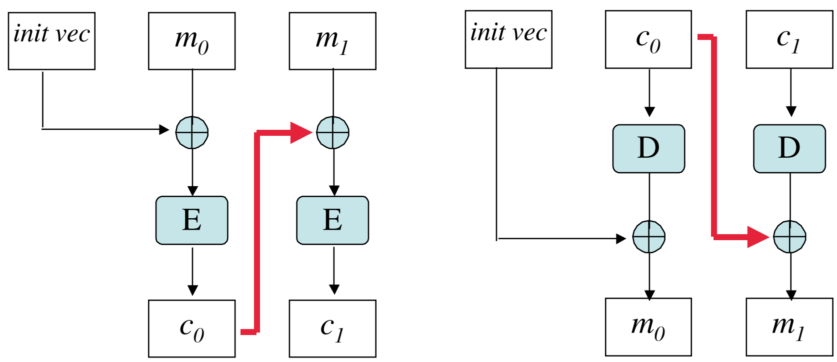
ESP - Encapsulating Security Header

# octets	
4	SPI (security parameter index)
4	sequence number
variable	IV (initialization vector)
variable	data
variable	padding
1	padding length
1	next header / protocol type
variable	authentication data

ESP Fields

- Same fields as in AH header:
  - SPI, sequence number, next header
- Initialization vector
  - Needed for some encryption algorithms
    - for example, when CBC mode is used (see next slide)
- Data: protected data, probably encrypted
- Padding: many 0's mainly in order to
  - make data be a multiple of a block size
    - Maybe required by adopted cryptographic algorithms
  - Or make [data, padding, padding length, next header] a multiple of four octets

CBC



Encipherment

Decipherment

(cont'd)

- Authentication data
  - Cryptographic integrity check
  - Zero length if ESP is providing only encryption

More on the Data Field in an ESP Header

- In Tunnel Mode
  - Begin at the IP header
- In Transport Mode
  - Begin at the IP payload
  - Begin at TCP header if a TCP payload

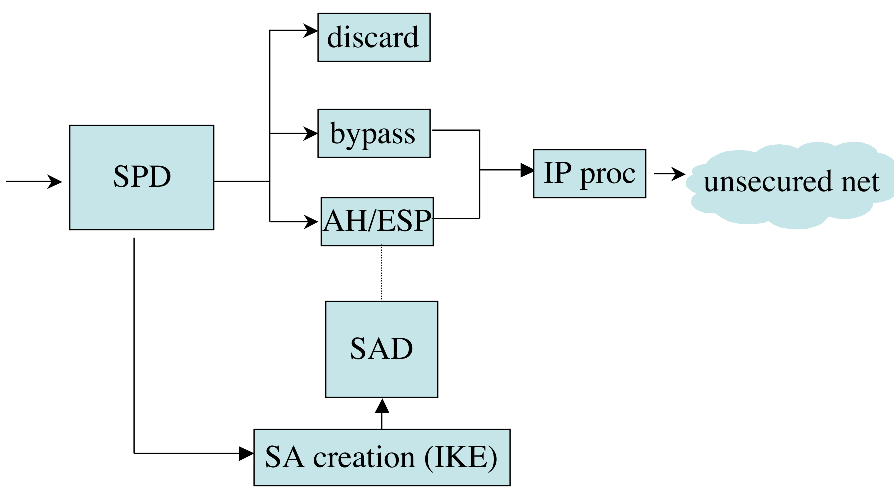
Security Policy Database

- An ordered list of SPD entries
- Each SPD entry specifies a policy: **applicability, disposition,** and **protection**
- Applicability: which packets are subject to policy
- Disposition: discard, bypass, or *apply IPsec*
- Protection: what kinds of SA to apply under this policy

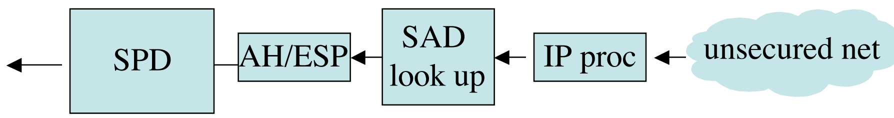
An Example of SPD entry

- Outbound SPD entry example:
  - IP: source=175.34.*.* destination=98.34.32.6
  - Protocol = 6 (TCP)
  - Port: source=any, destination=80
  - Disposition = IPsec
  - Protection = Details on what kind of SA to set up (e.g. ESP tunnel mode, DES, . . .)
- Similarly an inbound SPD entry can be defined

IPsec Outbound Traffic Processing



IPsec Inbound Traffic Processing



IPsec: IKE

Goal of IKE

- To do mutual authentication using long-term key
  - The long term key can be a public key
  - Or a pre-shared secret key
- And establish a session key

Three Pieces of IKE

- ISAKMP (Internet Security Association and Key Management Protocol)
  - RFC 2408
- IKE
  - RFC 2409
- DOI (Domain of Interpretation)
  - RFC 2407

Two Phases of IKE

- Phase 1: mutual authentication and session key establishment between Alice and Bob
  - Phase-1 exchange known as ISAKMP SA
  - Defined by ISAKMP (RFC 2408)
- Phase 2: multiple SAs between Alice and Bob
  - Phase-2 exchange creates IPSEC SA
  - Defined by IKE (RFC 2409?)

Why Two Phases?

- Multiple protocols
  - ISAKMP is not just for IPsec
- Multiple flows for Alice and Bob
  - Each needs a different SA

Phase 1 IKE

- Aggressive mode
  - Using 3 messages
- Main mode
  - Using 6 messages
  - And achieves additional functionalities
    - Hide endpoint id
    - Negotiate cryptographic algorithms
    - Etc.
- Both use Diffie-Hellman

Diffie-Hellman

- First public key cryptosystem
  - Still in use today
- Used to generate a **common** key by two users

Discrete Logarithm Problem

- Find a value of k such that

$$K = g^k \bmod p$$

for a given K, g, and prime p.

- Difficulty increases exponentially as p increases
- This is the basis of Diffie-Hellman

Algorithm

- All users share p and g
- Each user u chooses a private key $k(u)$ and a public key $K(u)$

$$K(u) = g^{k(u)} \bmod p$$

- When users A and B communicate,

$$\text{A: } s(\text{A}) = \text{E}_{k(\text{A})}(K(\text{B})) = K(\text{B})^{k(\text{A})} \bmod p$$

$$\text{B: } s(\text{B}) = \text{E}_{k(\text{B})}(K(\text{A})) = K(\text{A})^{k(\text{B})} \bmod p$$

s will be used as the secret key for A-B communication.

When A sends out a message encrypted with ***s***, only the one who holds (k(B), K(B)), which is B here, can decrypt**!**

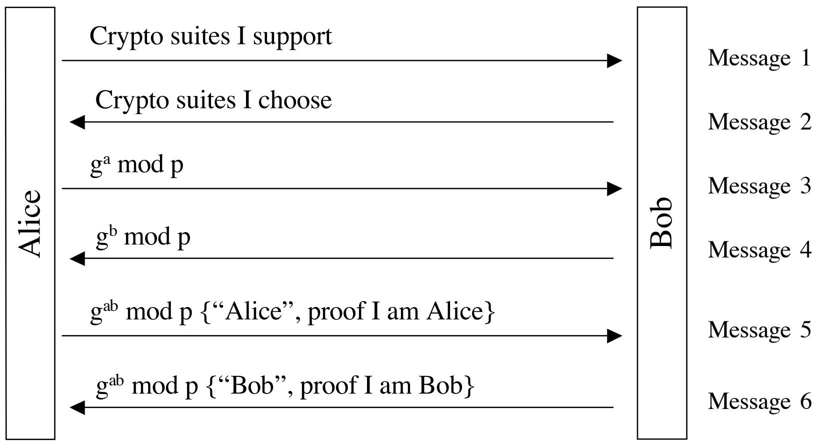
Example

- Alice and Bod chose $p = 53$, $g = 17$
- $k(\text{Alice}) = 5$, $k(\text{Bob}) = 7$
- $K(\text{Alice}) = 17^5 \bmod 53 = 40$
  $K(\text{Bob}) = 17^7 \bmod 53 = 6$
- Alice: $K(\text{Bob})^{k(\text{Alice})} \bmod \text{p} = 6^5 \bmod 53 = 38$
  Bob: $K(\text{Alice})^{k(\text{Bob})} \bmod \text{p} = 40^7 \bmod 53 = 38$

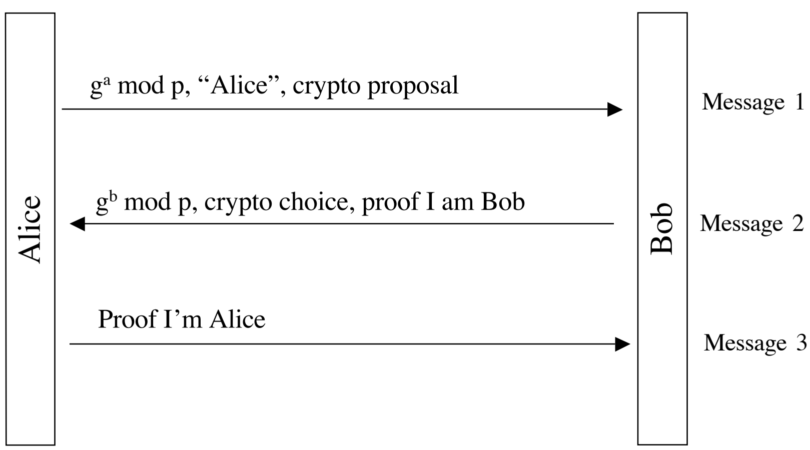
Diffie-Hellman Summary

- Based on the computational infeasibility to derive the private key of a public key
  - *p* must be very large (hundreds of bits)
- Diffie-Hellman is an example of **symmetric key exchange protocol**

Phase 1, Main Mode



Phase 1, Aggressive Mode



Negotiating Crypto Parameters

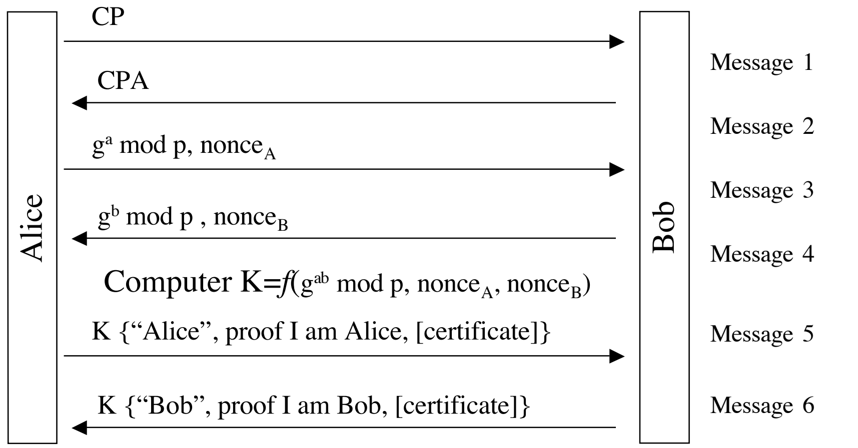
- Alice and Bob can negotiate various crypto methods
  - Encryption
  - Hash algorithm
  - Diffie Hellman parameters (only for main mode)
  - Authentication method
- Typically Alice provides an ordered list, and then Bob selects

Four Authentication Method

- Totally four authentication methods, depending on what type of long-term keys that Alice and Bob hold
  - Original public encryption key
  - Revised public encryption key
  - Public signature key
  - Pre-shared secret key

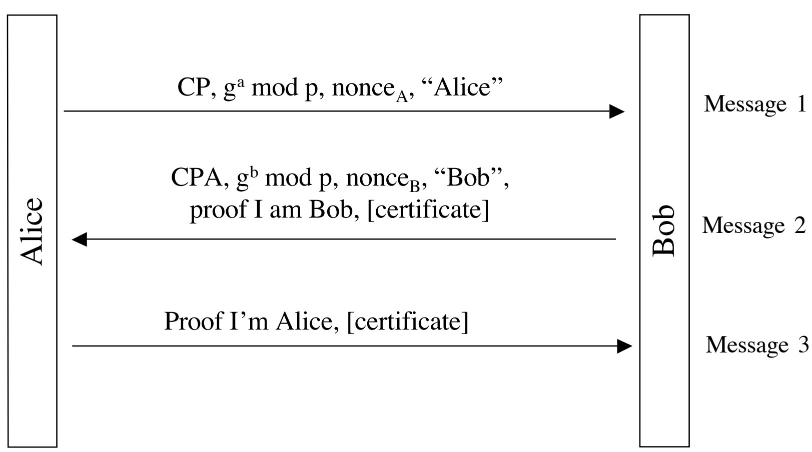
  - We focus on public key signature below
- Thus, totally eight variants of Phase 1 (remember it has two modes)

Phase 1, Main Mode
Public Signature Keys



CP: crypto proposal
CPA: CP accepted

Phase 1, Aggressive Mode
Public Signature Keys



Two Session Keys

- An integrity key
- An encryption key

- Used to protect some Phase 1 messages and ALL phase 2 IKE messages

Session Key Generation

- A pseudo random function
  - Hash result = prf (key, data)
  - Example: DEC CBC residue, or HMAC
- SKEYID = prf(nonces, g^{xy} mod p)
- SKEYID_d = prf(SKEYID, (g^{xy} mod p, cookies, 0))
- Integrity key (Kinc)
  - SKEYID_a = prf(SKEYID, (SKEYID_d, (g^{xy} mod p, cookies, 1)))
- Encryption key (Kenc)
  - SKEYID_e = prf(SKEYID,(SKEYID_a, (g^{xy} mod p, cookies, 2)))

Proof of Identity

- To prove the sender knows the key associated with the identity
  - E.g., Alice or Bob's private signature key
- IKE definition for the Proof "I'm Alice"
  - *prf*(SKEYID, (g^x, g^y, cookies, Alice's initial CP, Alice's identity))
- IKE definition for the Proof "I'm Bob"
  - *prf*(SKEYID, (g^y, g^x, cookies, Alice's initial CP, Bob's identity))

Phase-2 IKE: Setting Up IPSEC SAs

- Known as Quick Mode
- A 3-message protocol that negotiates parameters for the phase-2 SA
  - Crypto parameters
  - SPI (still remember what's this?)
- All messages are encrypted with K_{enc} and integrity protected with K_{int}

Phase 2, Quick Mode