NOTE: This lecture differs significantly from the textbook.

Network Security Solutions

Jun Li
lijun@cs.uoregon.edu

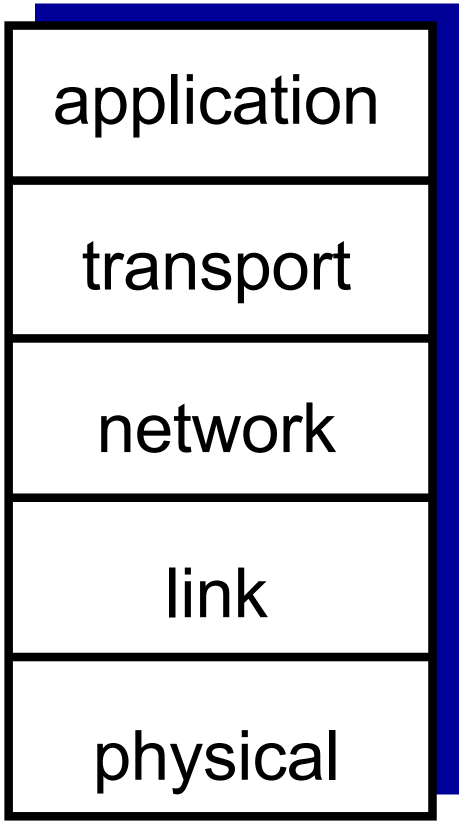
Learning Objectives

- Basic concepts
- Routing Security (Network Layer)
- IPSec (Network Layer)
- SSL/TLS (Transport Layer)
- Firewalls and Intrusion Detection Systems

Basic Concepts

Layered Network Security

- We focus on the Internet
- Security attacks toward the Internet can happen at each layer
- Let’s look at security defense at each layer
  - For example, what may happen at the physical layer?



application

transport

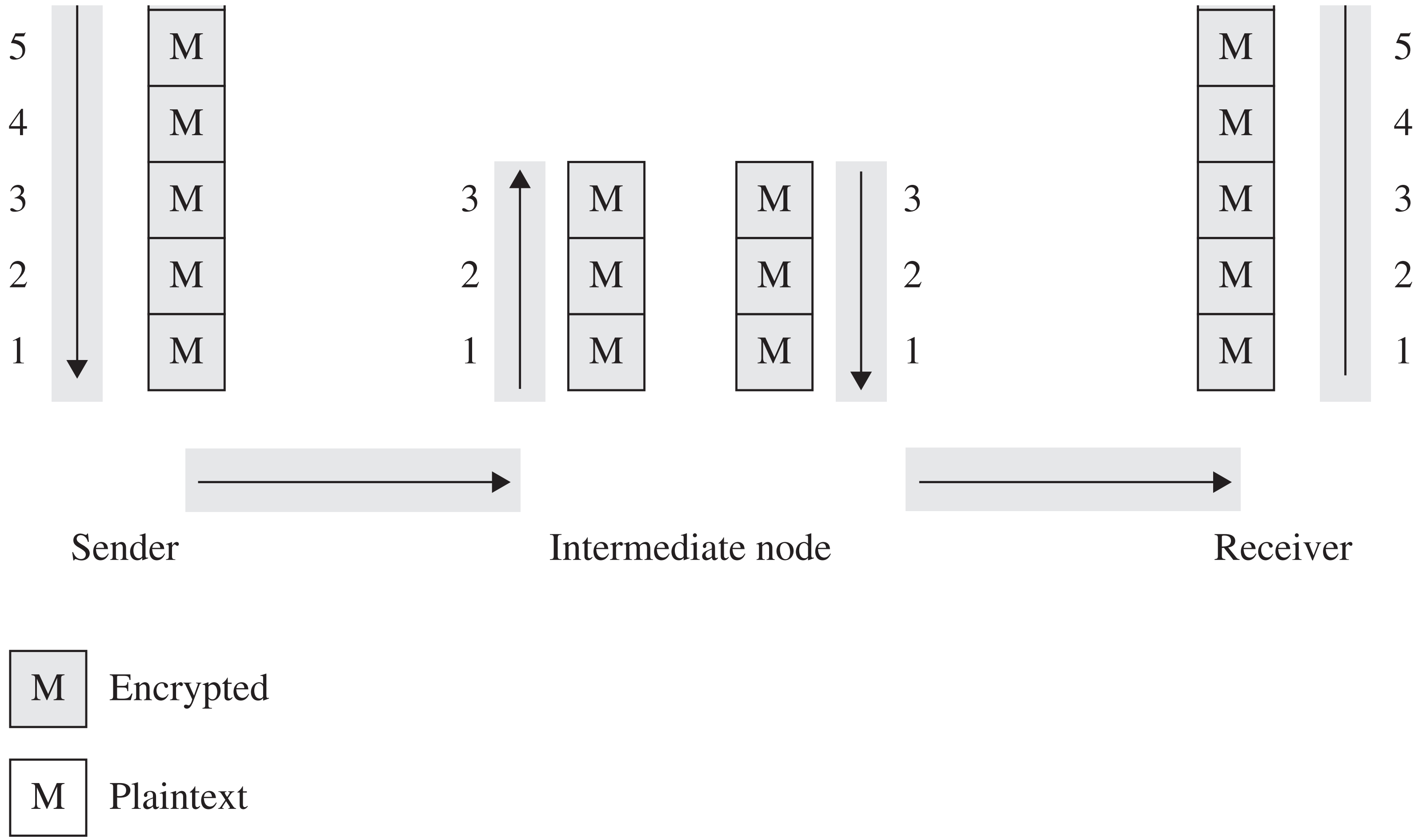
network

link

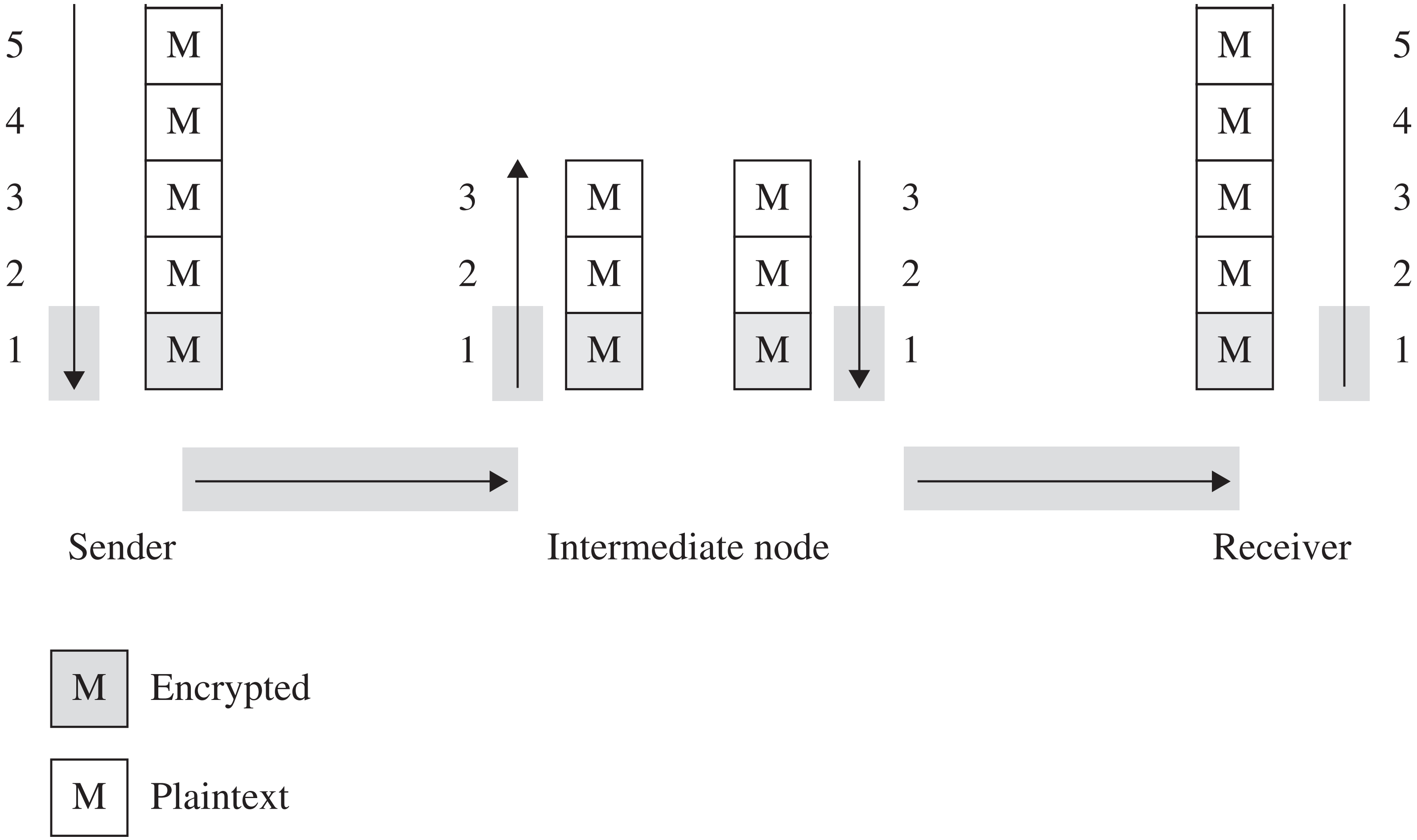
physical

Link Layer

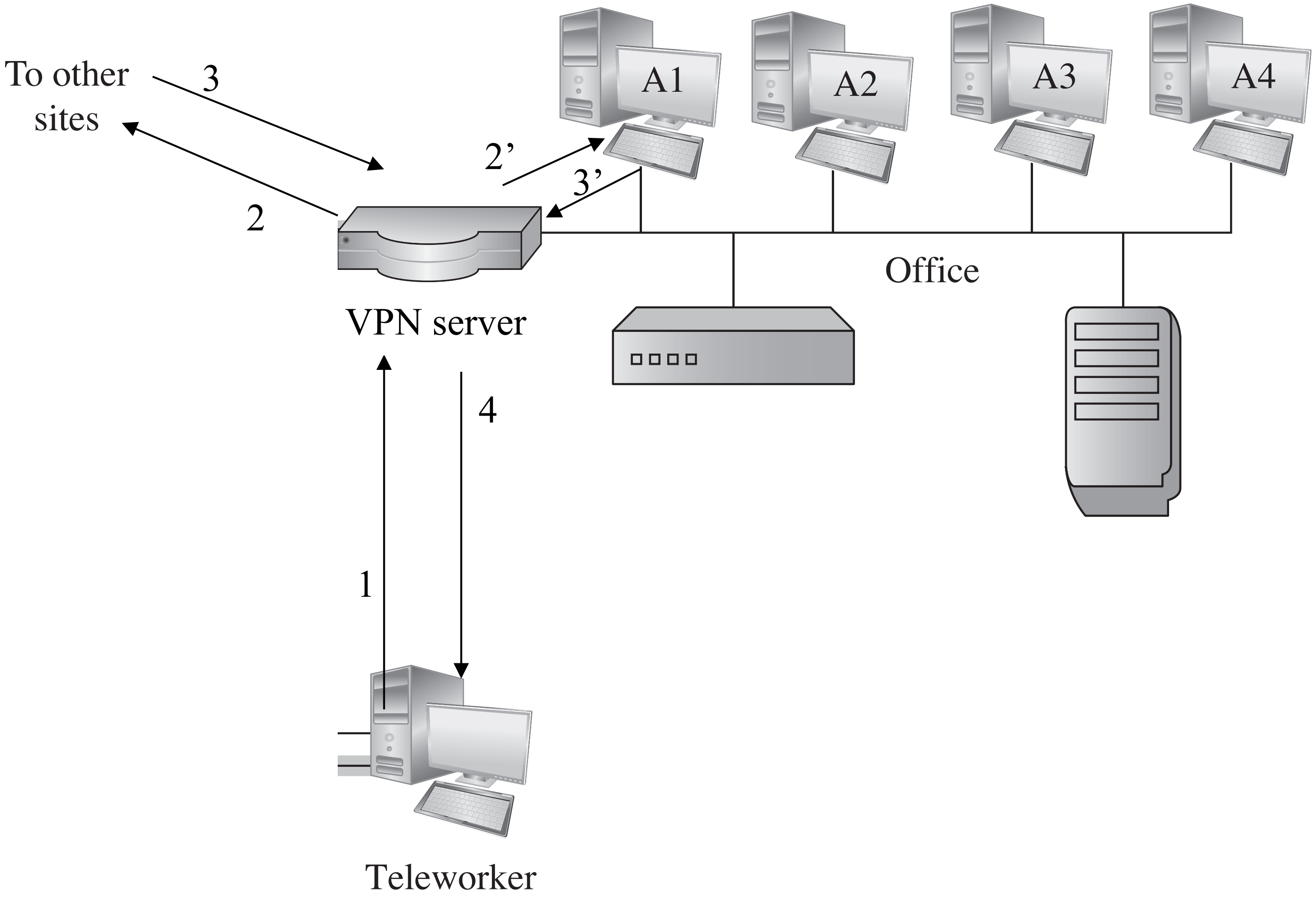
End-to-End Encryption



Link Encryption



VPN


To other sites
3
2
2'
3'
A1
A2
A3
A4
Office
VPN server
4
1
Teleworker

Network Layer

Network Layer Security

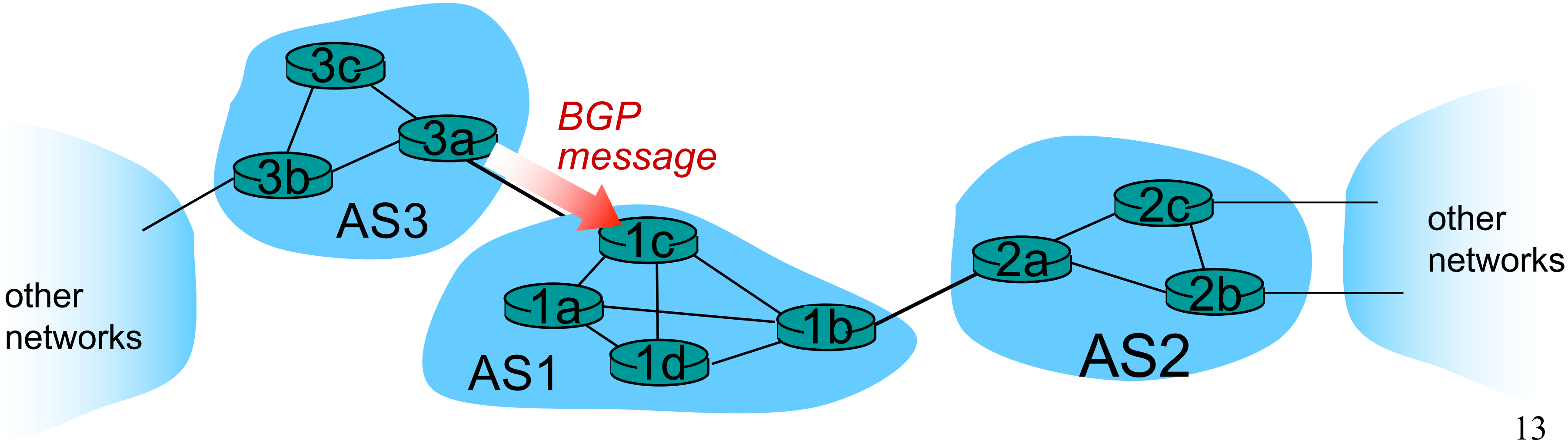
- Routing Security: control plane security
- IPSec: data plane security

Routing Security

BGP basics

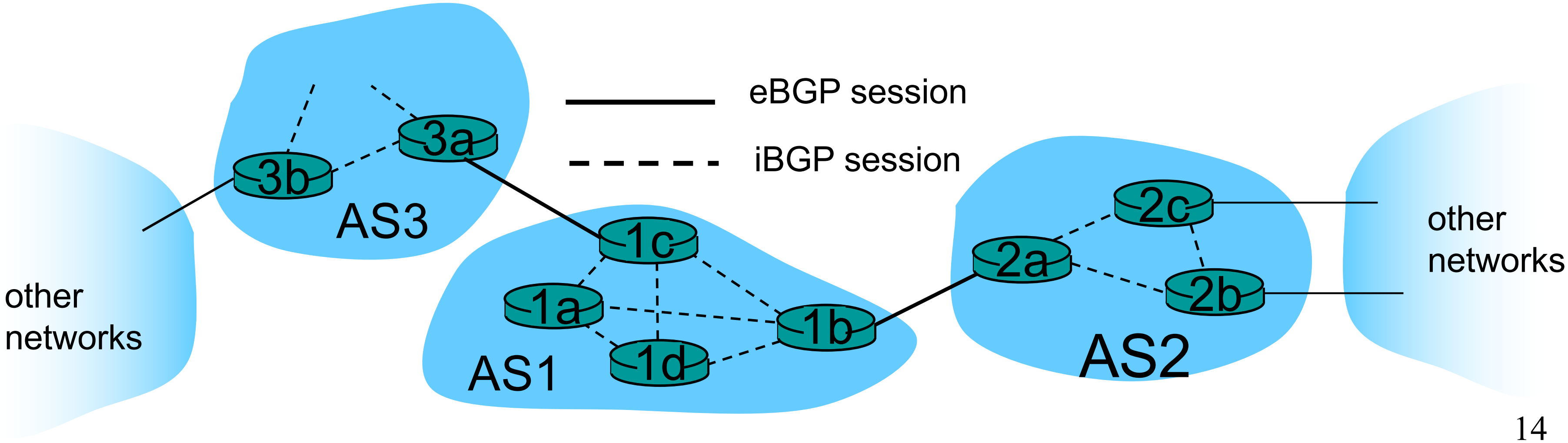
- BGP session: two BGP routers ("peers") exchange BGP messages:
  - advertising *paths* to different destination network prefixes ("path vector" protocol)
  - exchanged over semi-permanent TCP connections

- when AS3 advertises a prefix to AS1:
  - AS3 *promises* it will forward datagrams towards that prefix
  - AS3 can aggregate prefixes in its advertisement



BGP basics: distributing path information

- using eBGP session between 3a and 1c, AS3 sends prefix reachability info to AS1.
  - 1c can then use iBGP to distribute new prefix info to all routers in AS1
  - 1b can then re-advertise new reachability info to AS2 over 1b-to-2a eBGP session
- when router learns of new prefix, it creates entry for prefix in its forwarding table.



Routing Attacks

- Internet routing is not secure
  - Routers trust each other?
  - Many routing attacks have happened
- Origin
  - Blackhole attack
- Path
  - Prefix hijacking
  - Route leaks

How to Secure Routing?

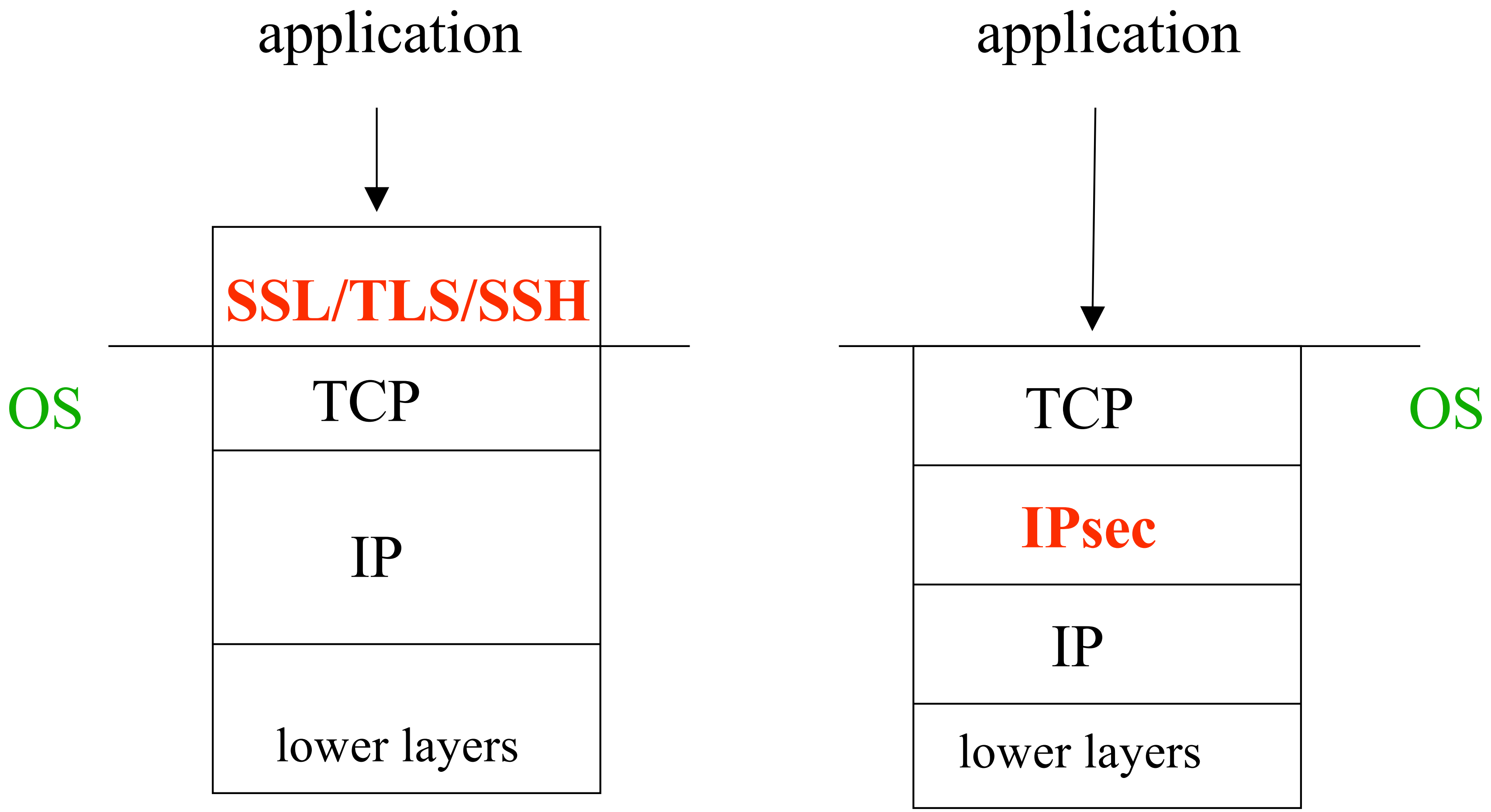
- Origin Authentication
  - Sign who you are
- Path Authentication
  - Sign your attestation: I have seen this path.
  - {A, sig_by_A}
  - {B, {A, sig(A)}}, sig_by_B}
  - {C, {B, {A, sig(A)}}, sig_by_B}, sig_by_C
  - …

IPsec

IPsec as a Real-Time Protocol

- A real-time protocol is one where parties negotiate interactively to authenticate 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



application

SSL/TLS/SSH

OS

TCP

IP

lower layers

application

TCP

OS

IPsec

IP

lower layers

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 notify the TCP layer if newly received data is bogus
- 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 set up a secure channel
  - Called **Security Association**
- Then rely on IPsec to protect the channel

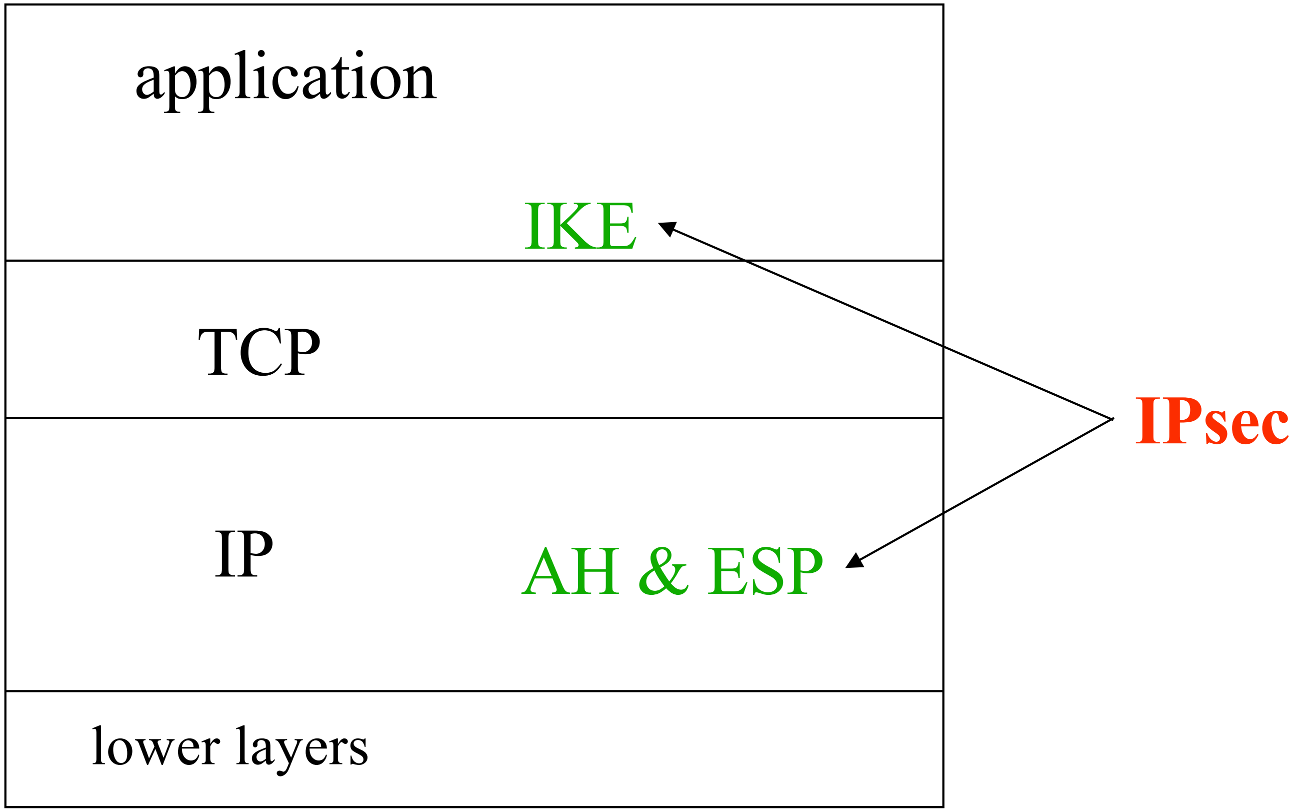
What does IPsec Accomplish?

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

- Similar to setting 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


application
IKE
TCP
IPsec
IP
AH & ESP
lower layers

IPsec Deployment

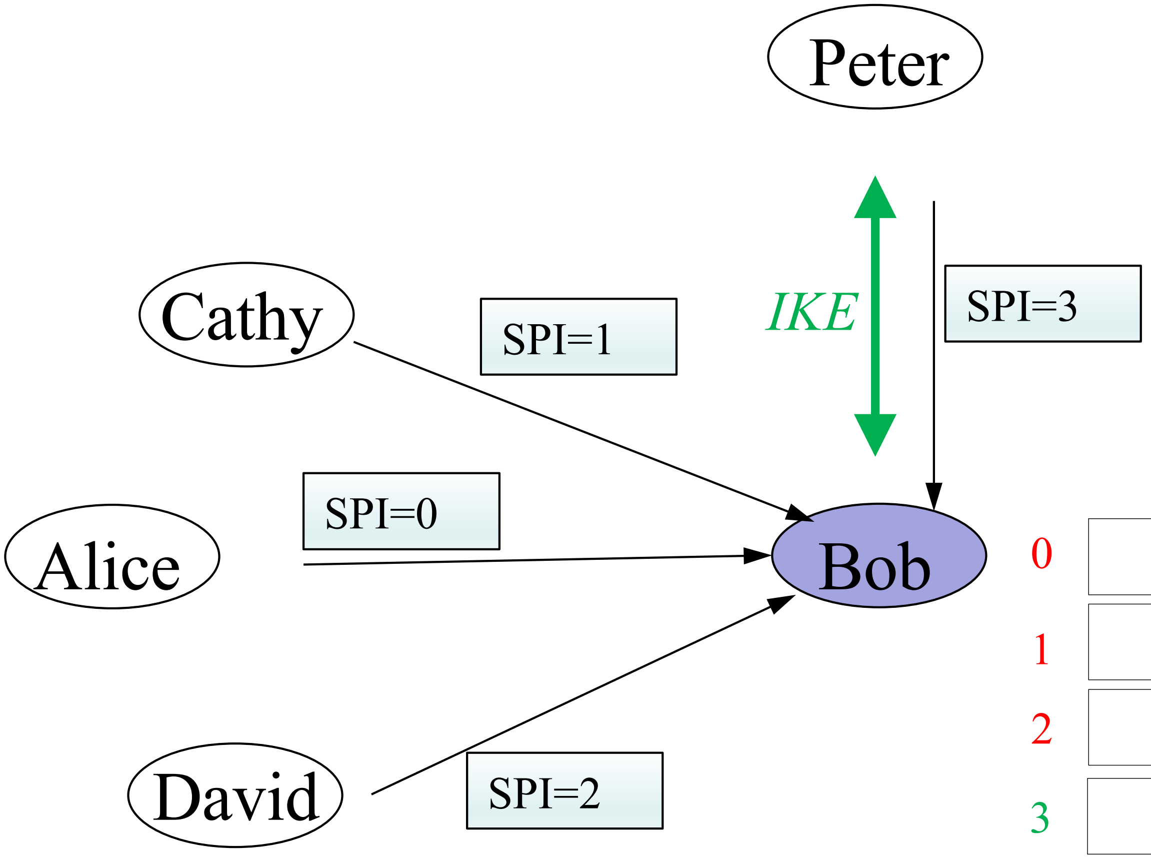
- Individual host: an end system can implement its own protection end-to-end or hop-by-hop
- Host community: a single security gateway (e.g. a firewall) can protect an entire domain of hosts
- Pairings: host-to-host, host-to-gateway, gateway-to-gateway
  - Or combined

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

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



Peter

IKE

SPI=3

Cathy

SPI=1

SPI=0

Alice

Bob

David

SPI=2

0	SA w/ Alice
1	SA w/ Cathy
2	SA w/ David
3	SA w/ Peter

SAD

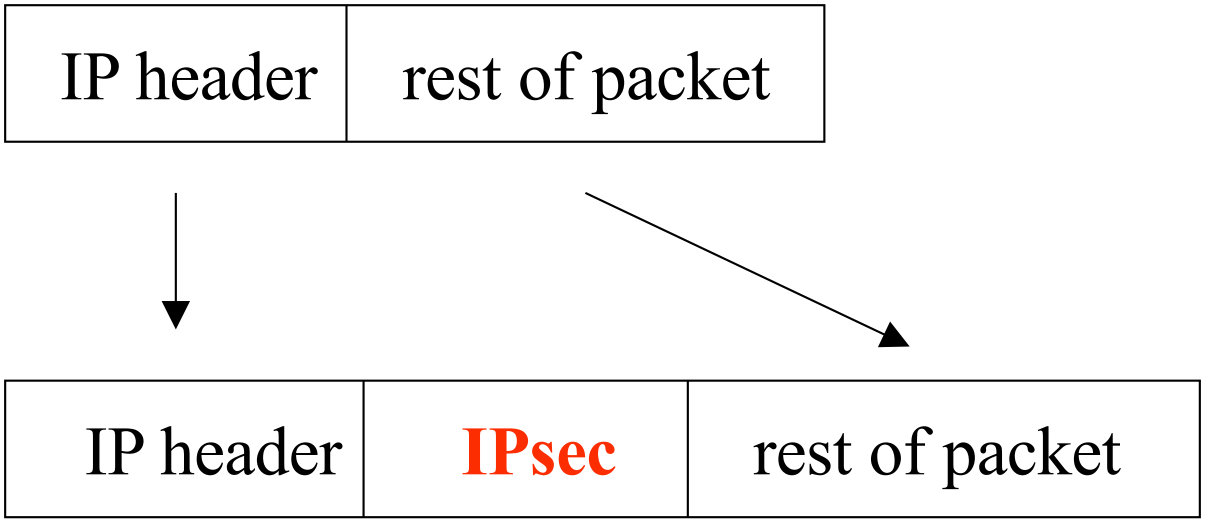
AH & ESP

- 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

Two IPsec Modes

- Transport mode
- Tunnel mode

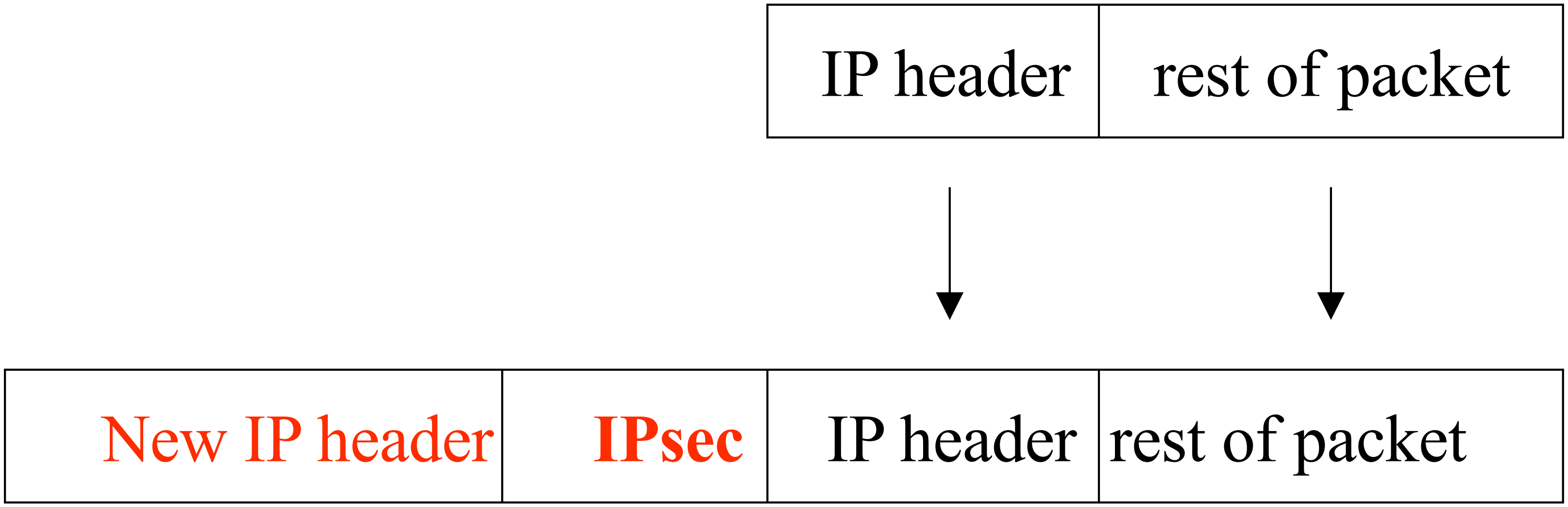
Transport Mode



IP header | rest of packet

IP header | **IPsec** | rest of packet

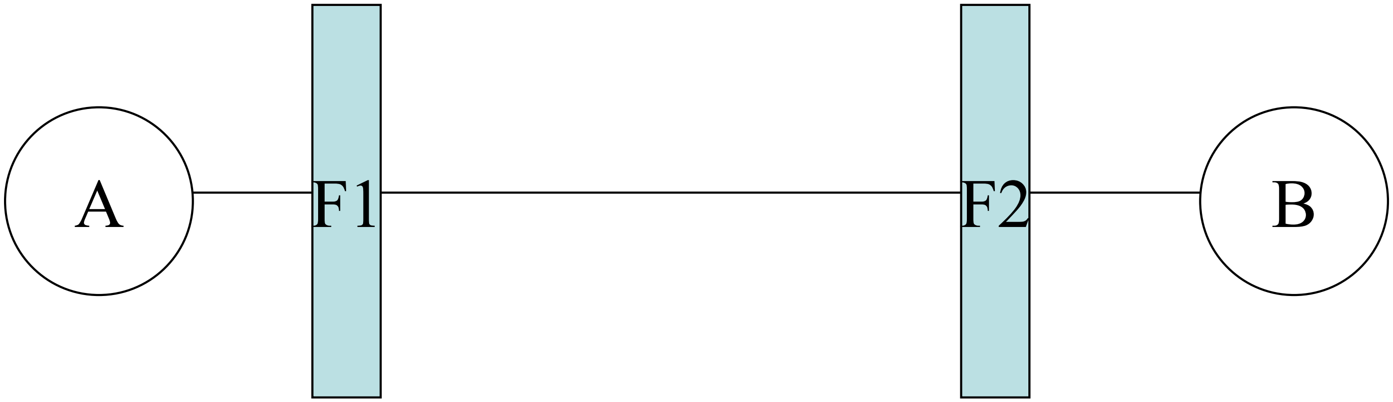
Tunnel Mode


IP header
rest of packet
New IP header
IPsec
IP header
rest of packet

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



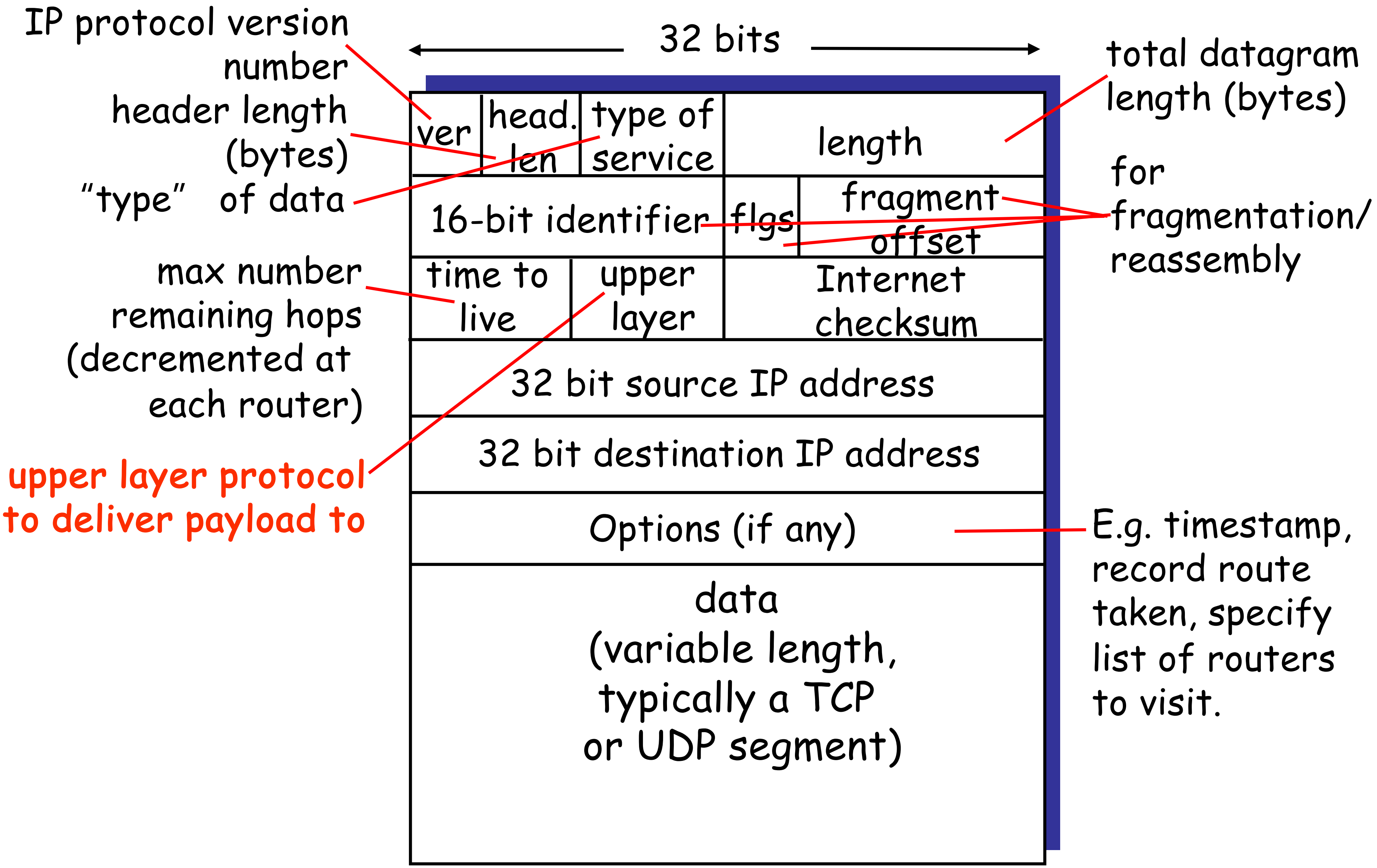
IP: src=F1, dst=F2	ESP	IP: src=A, dst=B	

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



IP protocol version number

header length (bytes)

"type" of data

max number remaining hops (decremented at each router)

upper layer protocol to deliver payload to

32 bits

ver | head. len | type of service | length

16-bit identifier | flgs | fragment offset

time to live | upper layer | Internet checksum

32 bit source IP address

32 bit destination IP address

Options (if any)

data (variable length, typically a TCP or UDP segment)

total datagram length (bytes)

for fragmentation/ reassembly

E.g. timestamp, record route taken, specify list of routers to visit.

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

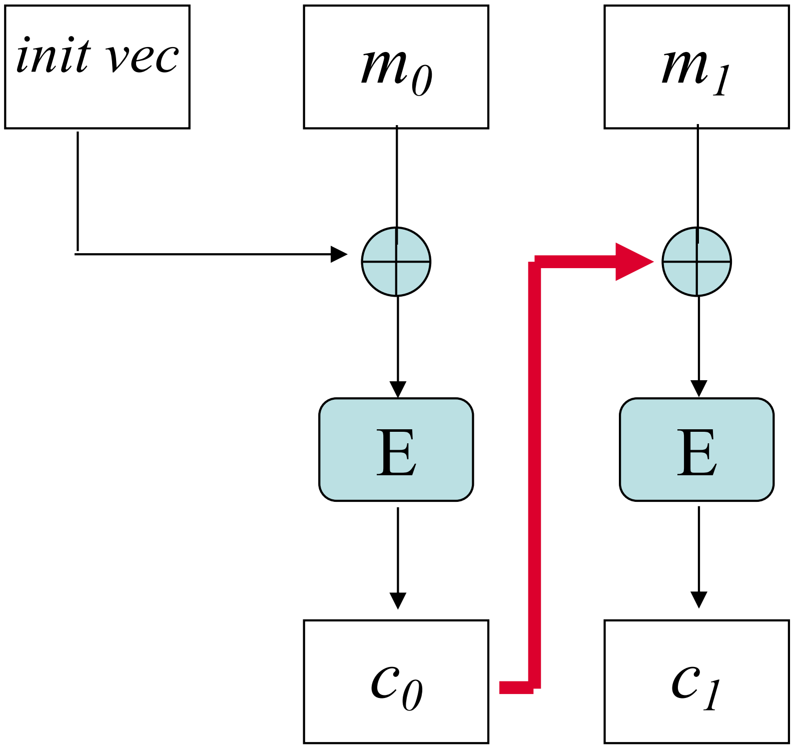
ESP - Encapsulating Security Header

# 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

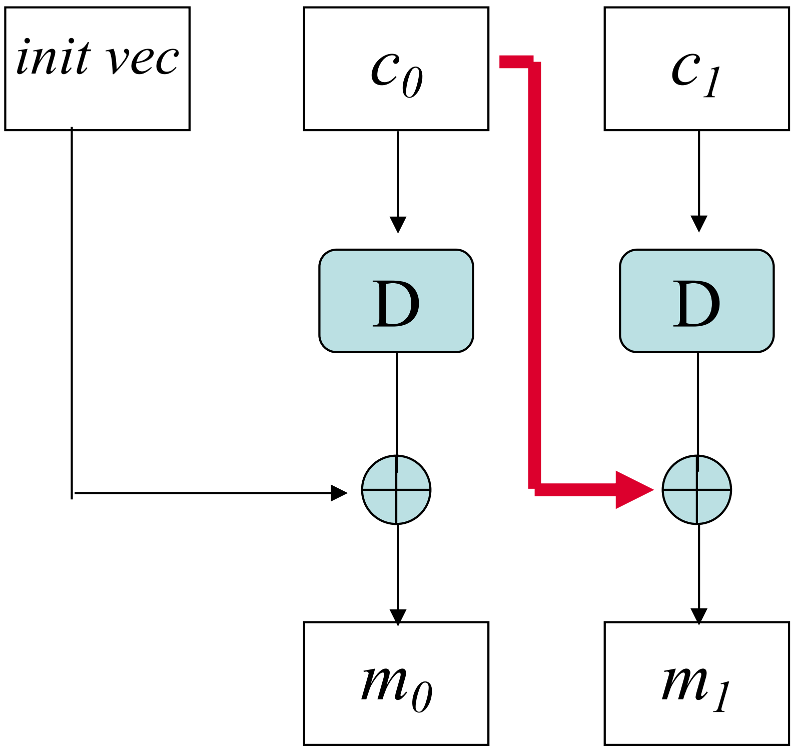
ESP Fields

- 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

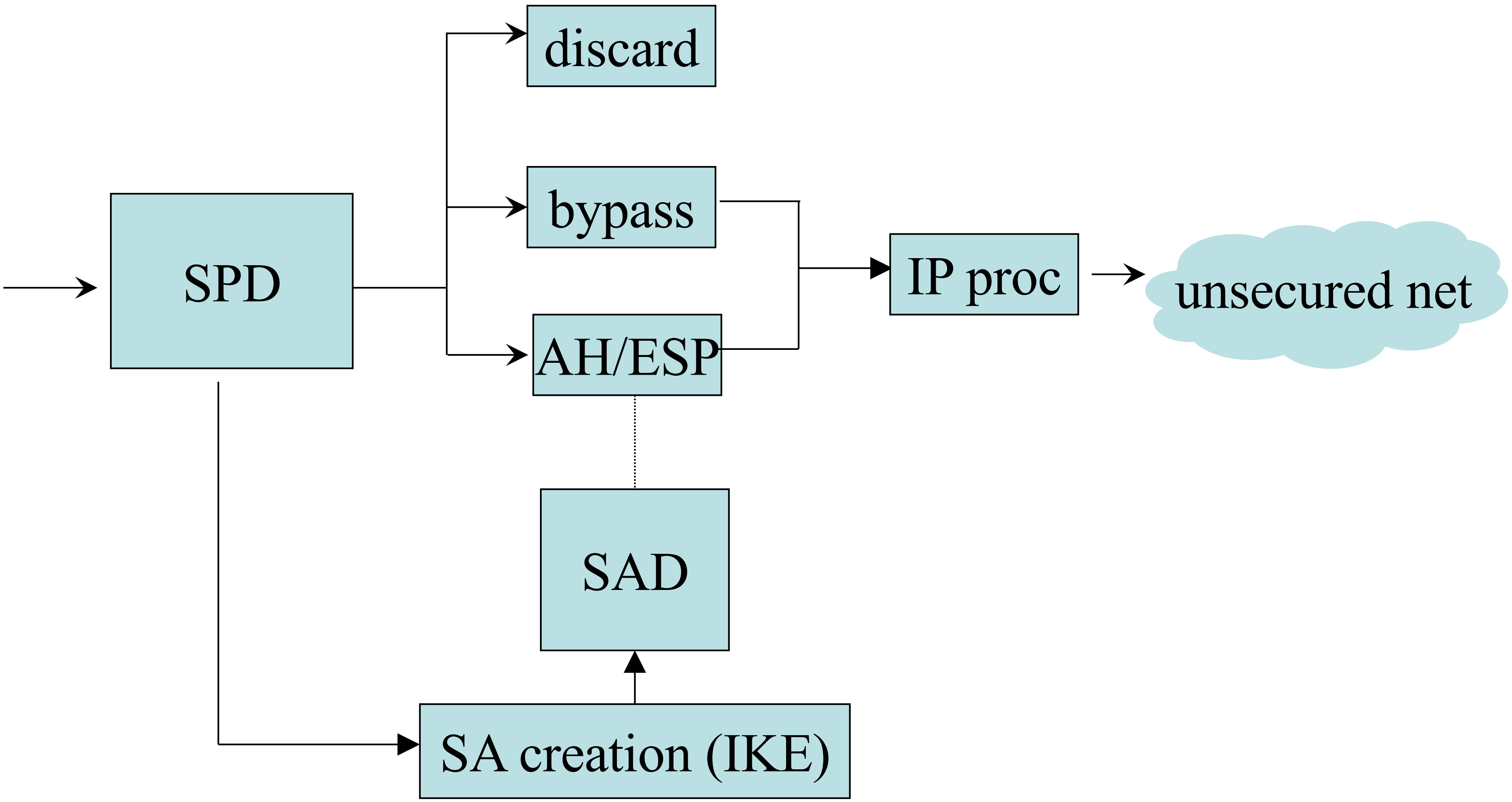
Security Policy Database

- 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

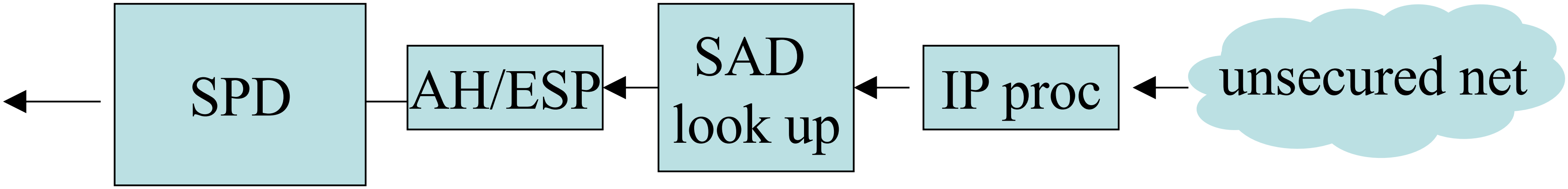
An Example of SPD entry

- 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

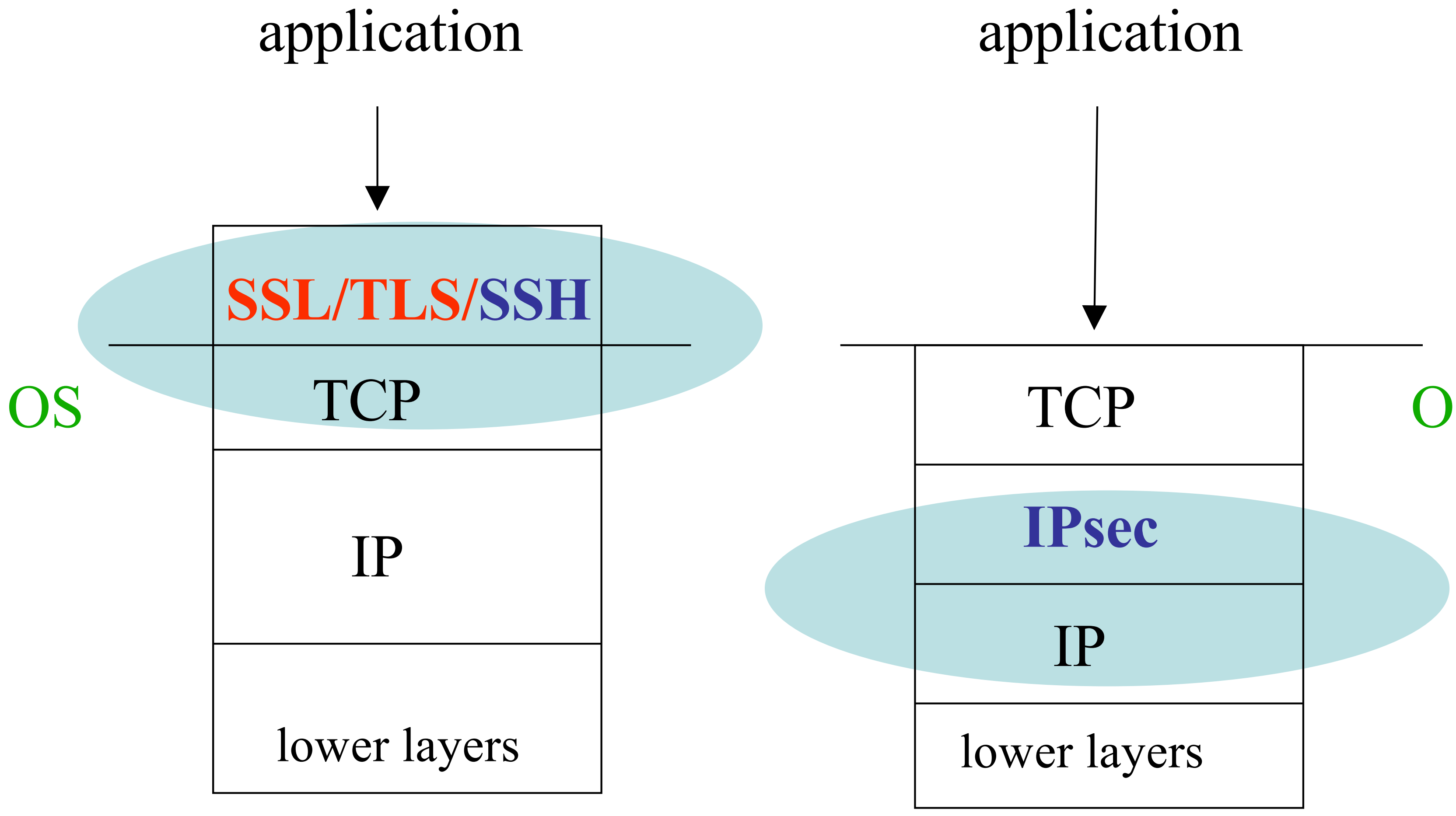


SSL/TLS

SSL/TLS as Real-Time Protocols

- A real-time protocol is one where parties negotiate interactively to authenticate each other and establish a session key
- Examples: IPsec, SSL/TLS, SSH
  - Public key based
- SSL: Secure Socket Layer
- TLS: Transport Layer Security

Security at Layer 4 vs. 3



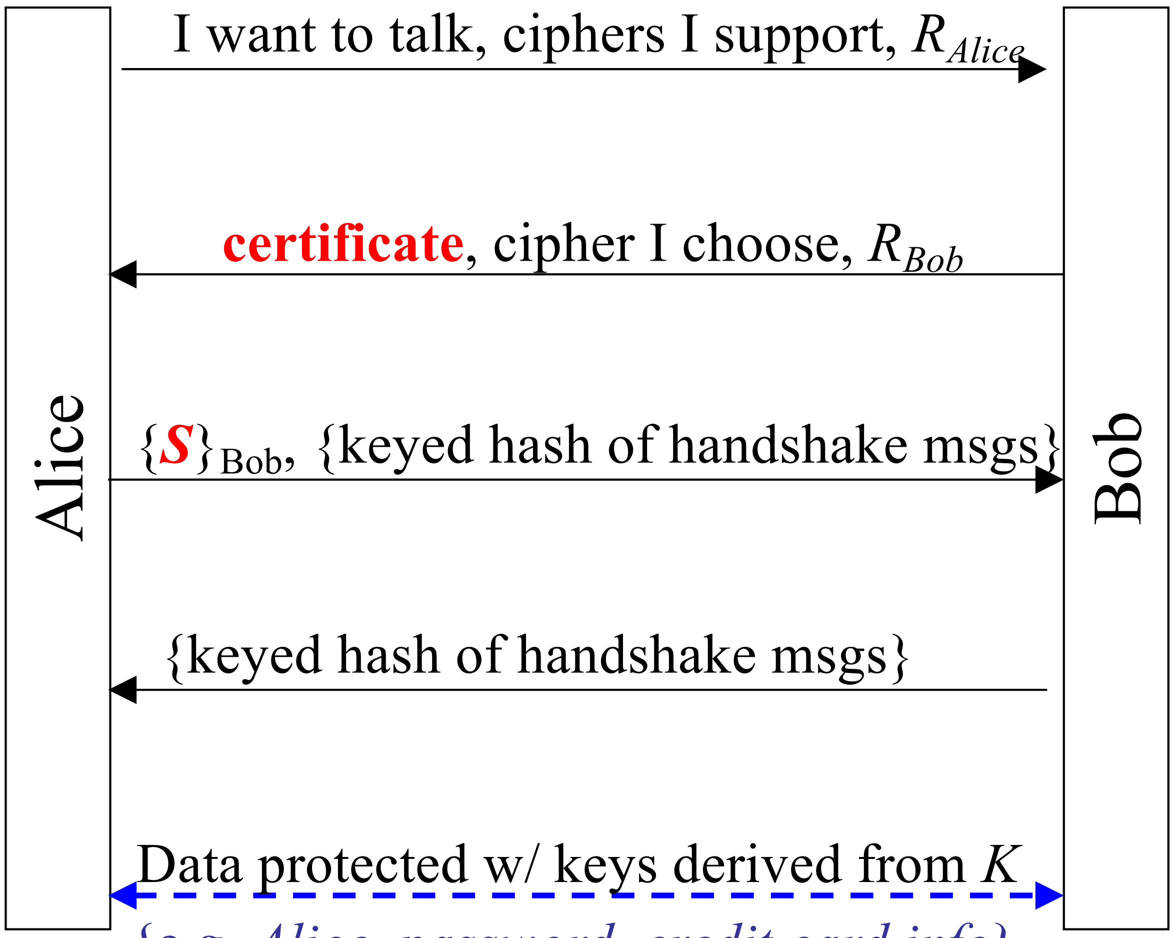
Assumption: TCP/IP are in the OS

Quick History

- SSLv1: never deployed
- SSLv2: deployed in Netscape Navigator 1.1 in 1995
- Microsoft introduced PCT (Private Communication Tech) by improving SSLv2
- Netscape overhauled the protocol as SSLv3
- IETF introduced TLS to unify all of them
  - Currently TLS v1.3

Basic Protocol

Choose secret $\boldsymbol{S}$, compute $K=f(\boldsymbol{S}, R_{Alice}, R_{bob})$



Alice

I want to talk, ciphers I support, R_{Alice}

certificate, cipher I choose, R_{Bob}

$\{\boldsymbol{S}\}_{\text{Bob}}$, {keyed hash of handshake msgs}

{keyed hash of handshake msgs}

Data protected w/ keys derived from K

Bob

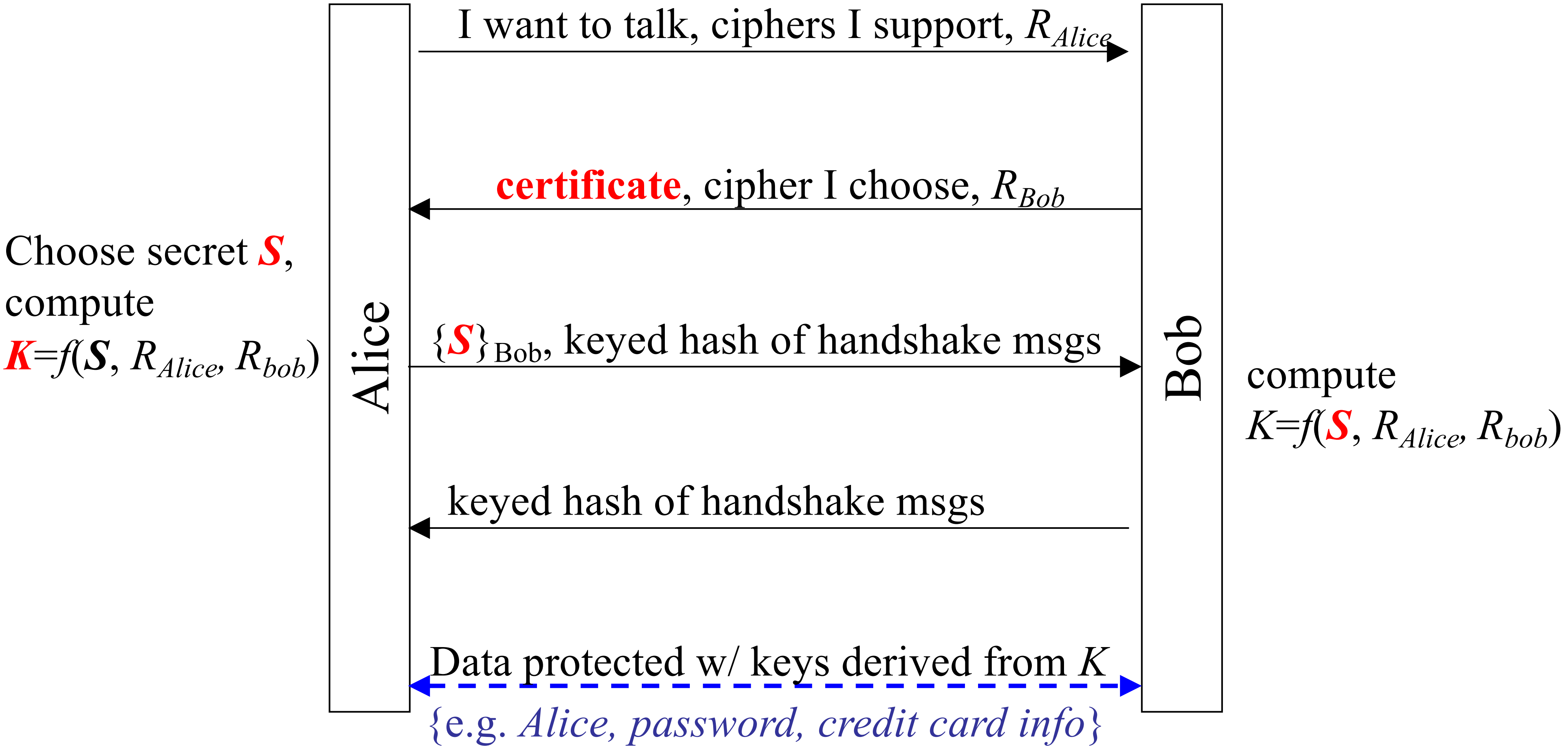
compute $K=f(\boldsymbol{S}, R_{Alice}, R_{bob})$

{e.g. *Alice, password, credit card info*}

Several Important Terms

- $\boldsymbol{R_{Alice}}$: a random number from Alice
- ***S***: pre-master secret
- ***K***: master secret
- $\{\}_{\mathrm{Bob}}$ stands for message encrypted with Bob' s public key
- {} stands for **protected** message using encryption and/or integrity protection through secret key algorithm

If a Keyed Hash Result in ***Plaintext***



Choose secret ***S***, compute

$K=f(S, R_{Alice}, R_{bob})$

Alice

I want to talk, ciphers I support, R_{Alice}

certificate, cipher I choose, R_{Bob}

$\{S\}_{\mathrm{Bob}}$, keyed hash of handshake msgs

keyed hash of handshake msgs

Data protected w/ keys derived from *K*

{e.g. *Alice, password, credit card info*}

Bob

compute

$K=f(S, R_{Alice}, R_{bob})$

How Bob Verifies the Keyed Hash

- Decrypt $\{\boldsymbol{S}\}_{\text{Bob}}$ using his private key
- Compute $\boldsymbol{K}=f(\mathbf{S}, R_{Alice}, R_{Bob})$
- Calculate *hash*($\boldsymbol{K}$, (m1, m2, “CLNT”))
  - HMAC algorithm
- Compares the result with the received one
- Verified if equal

- Q: must the keyed hash be protected?

How Alice Verifies the Key Hash

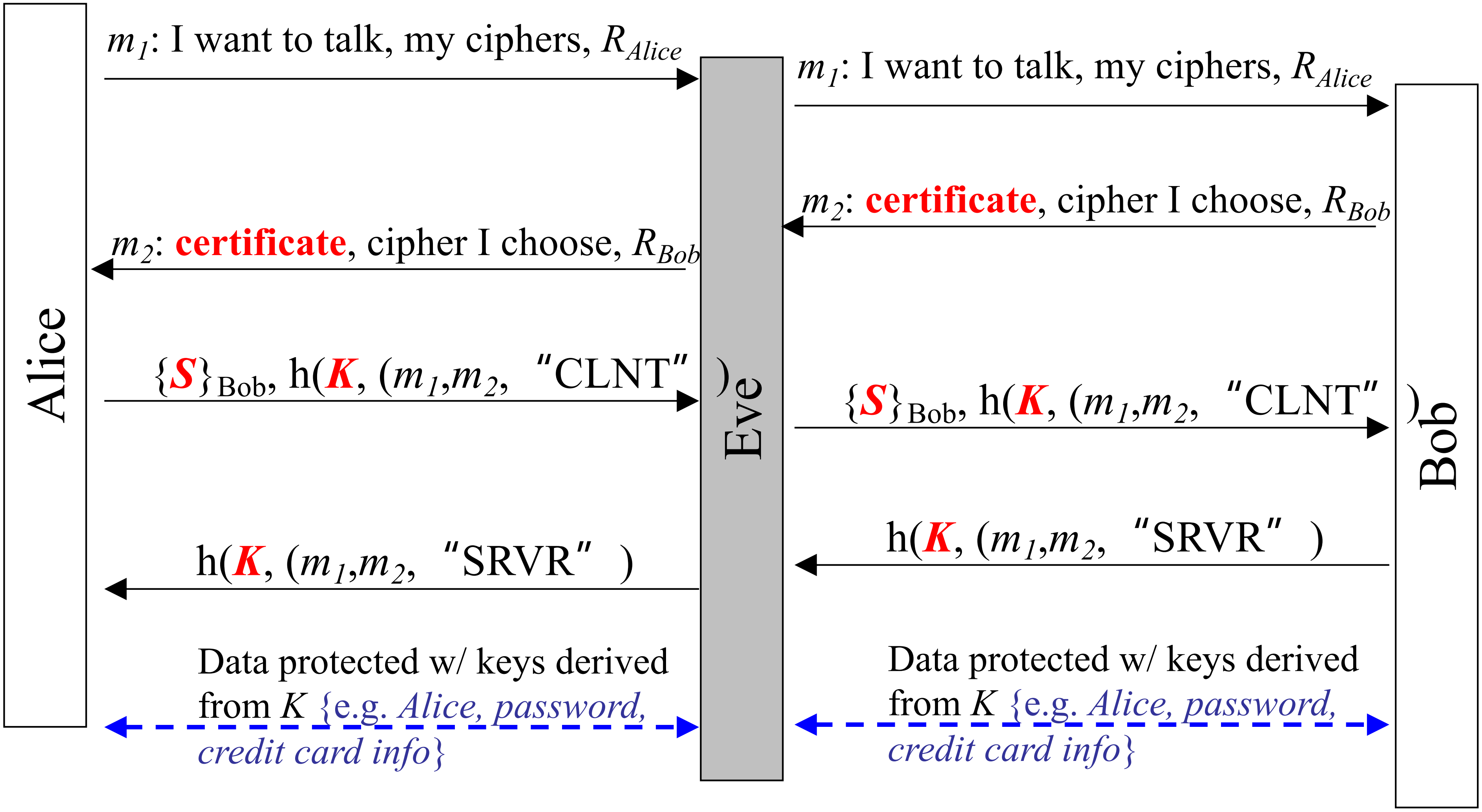
- Calculate *hash*(***K***, (m1, m2, “SRVR”))
  - HMAC algorithm
  - Recall Alice knows ***K*** already
  - The constant string make the hash different from what Bob receives
- Compares the result with the received one
- Verified if equal

- Q: must the keyed hash be protected?

Questions

- Can Eve eavesdrop?
- Can Mallury manipulate the data stream?

When Eve is Eavesdropping



Alice

Eve

Bob

m_1: I want to talk, my ciphers, R_{Alice}

m_1: I want to talk, my ciphers, R_{Alice}

m_2: **certificate**, cipher I choose, R_{Bob}

m_2: **certificate**, cipher I choose, R_{Bob}

$\{\boldsymbol{S}\}_{\text{Bob}}$, h($\boldsymbol{K}$, ($m_1$,$m_2$, "CLNT")

$\{\boldsymbol{S}\}_{\text{Bob}}$, h($\boldsymbol{K}$, ($m_1$,$m_2$, "CLNT")

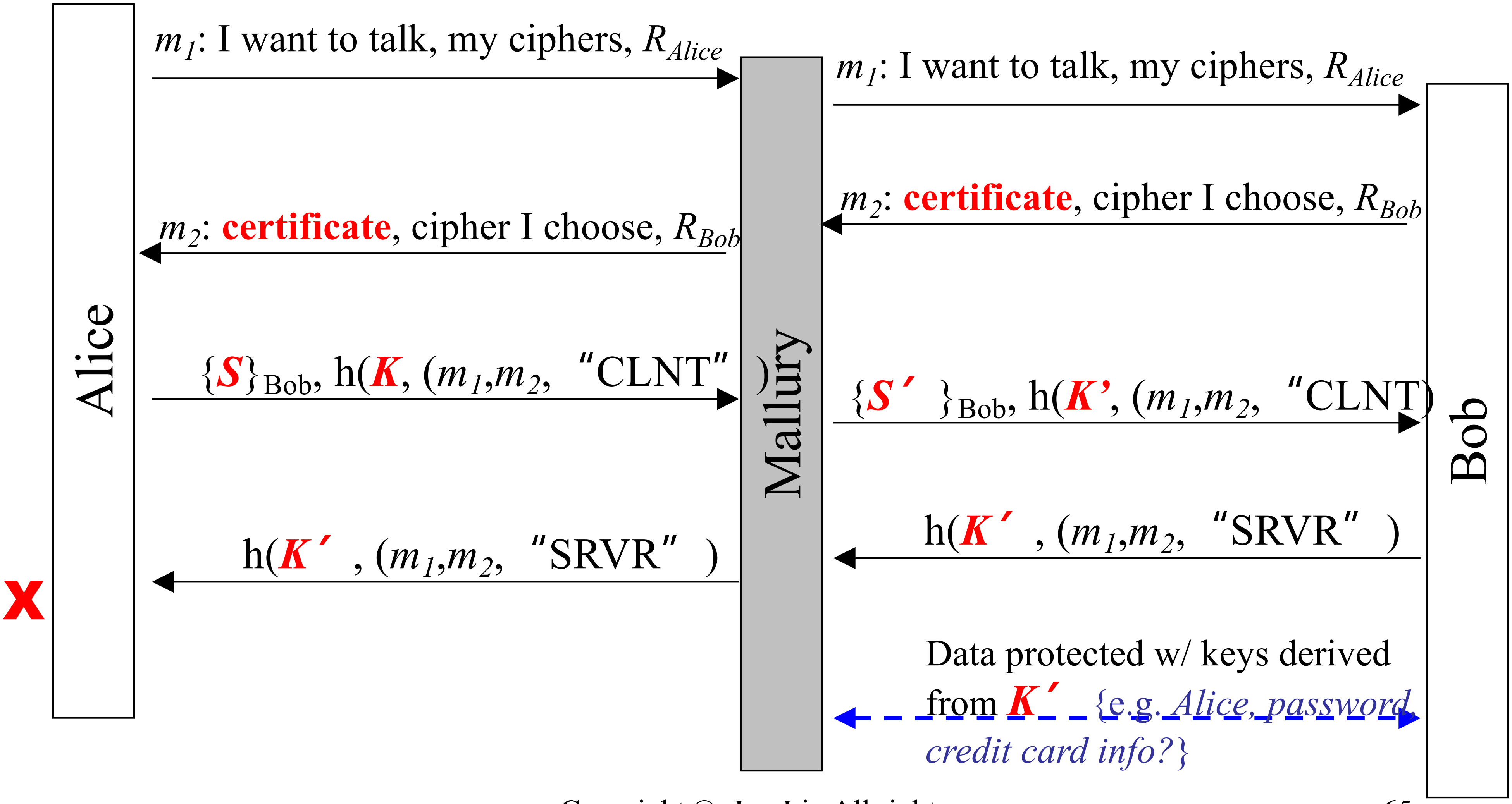
h($\boldsymbol{K}$, (m_1,m_2, "SRVR")

h($\boldsymbol{K}$, (m_1,m_2, "SRVR")

Data protected w/ keys derived from K {e.g. *Alice, password, credit card info*}

Data protected w/ keys derived from K {e.g. *Alice, password, credit card info*}

When Mallury is Manipulating



Alice

Mallury

Bob

m_1: I want to talk, my ciphers, R_{Alice}

m_1: I want to talk, my ciphers, R_{Alice}

m_2: **certificate**, cipher I choose, R_{Bob}

m_2: **certificate**, cipher I choose, R_{Bob}

$\{S\}_{Bob}$, h(K, (m_1,m_2, "CLNT")

$\{S'\}_{Bob}$, h(K', (m_1,m_2, "CLNT)

h(K' , (m_1,m_2, "SRVR")

h(K' , (m_1,m_2, "SRVR")

x

Data protected w/ keys derived from K' *{e.g. Alice, password, credit card info?}*

Questions

- When hashing, why add “CLNT” or “SRVR” ?
- What if not?

If Verified, What does Bob Prove?

- The following can be regarded as the same entity:
  - The one sending, or forwarding, message 1
  - the one computing the pre-master secret that Bob received
  - the one sending message 3
- But not necessarily Alice, even claimed so!
  - Could be Mallury!
  - But Alice won't be deceived

If Verified, What does Alice Prove?

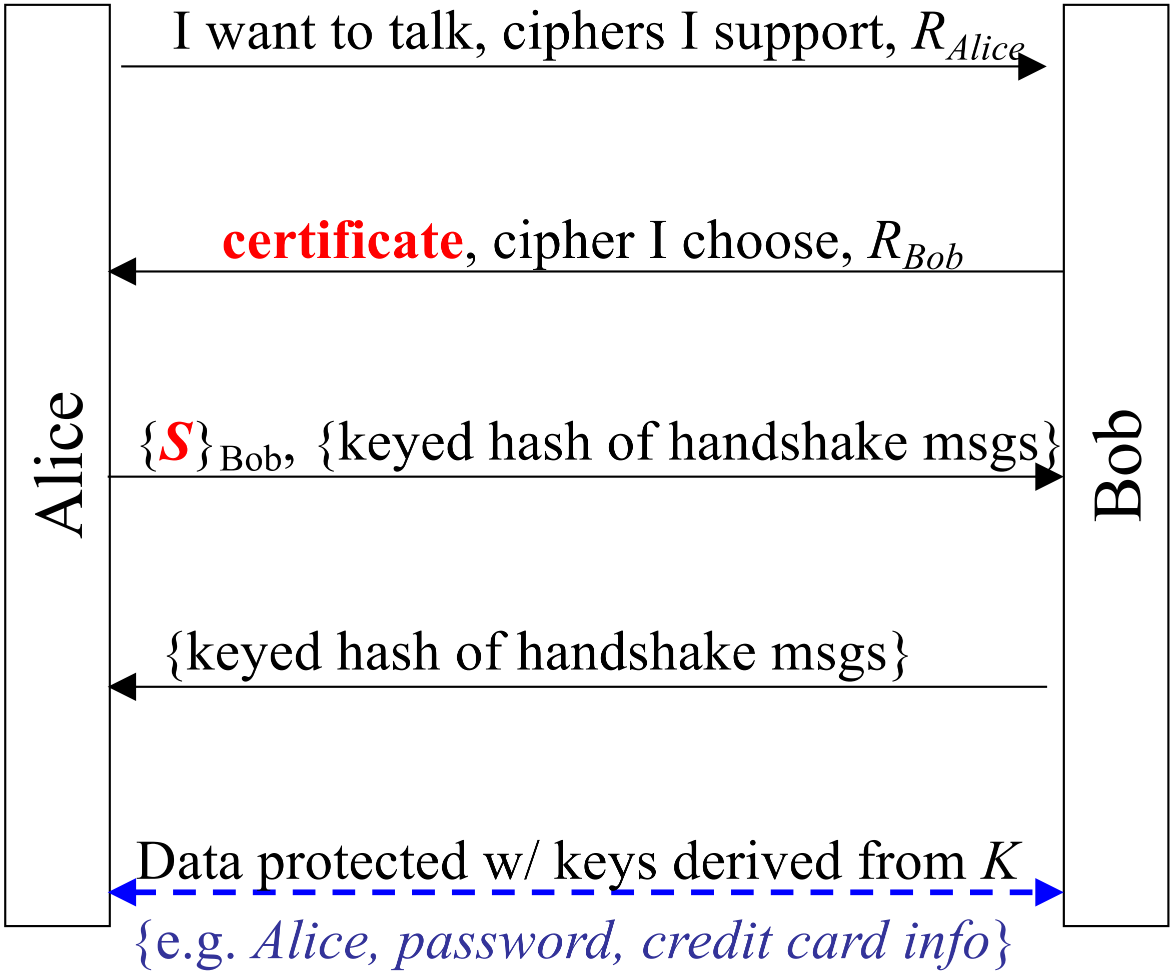
- The following are the same entity:
  - The one sending message 2
  - the one computing ***S*** and ***K*** on the other end, and
  - the one sending message 4
- And this entity is Bob!
  - Based on the certificate
- Also, this entity knows ***S*** and ***K***
  - ***S and K are decided by Alice***
- All handshake messages so far have NOT been tampered
  - Otherwise?

More on SSL/TLS

- Six secrets to protect Alice-Bob communication
- Handling a long *session* with many *connections*
- What if Alice also has a certificate

Six Secrets

- In fact, it's not a single key ***K*** for a session
- Definition: write keys and read keys
  - Write keys: keys for transmission
  - Read keys: keys for reception
- Each direction needs three write keys
  - Integrity protection key
  - Encryption key
  - IV, if required by encryption algorithms
- And also three read keys
- Computed using $g_i(K, R_{Alice}, R_{Bob})$



Alice

Bob

I want to talk, ciphers I support, R_{Alice}

certificate, cipher I choose, R_{Bob}

Choose secret ***S***, compute $K = f(\boldsymbol{S}, R_{Alice}, R_{bob})$

$\{\boldsymbol{S}\}_{\text{Bob}}$, {keyed hash of handshake msgs}

compute $K = f(\boldsymbol{S}, R_{Alice}, R_{bob})$

compute g_i

{keyed hash of handshake msgs}

compute g_i

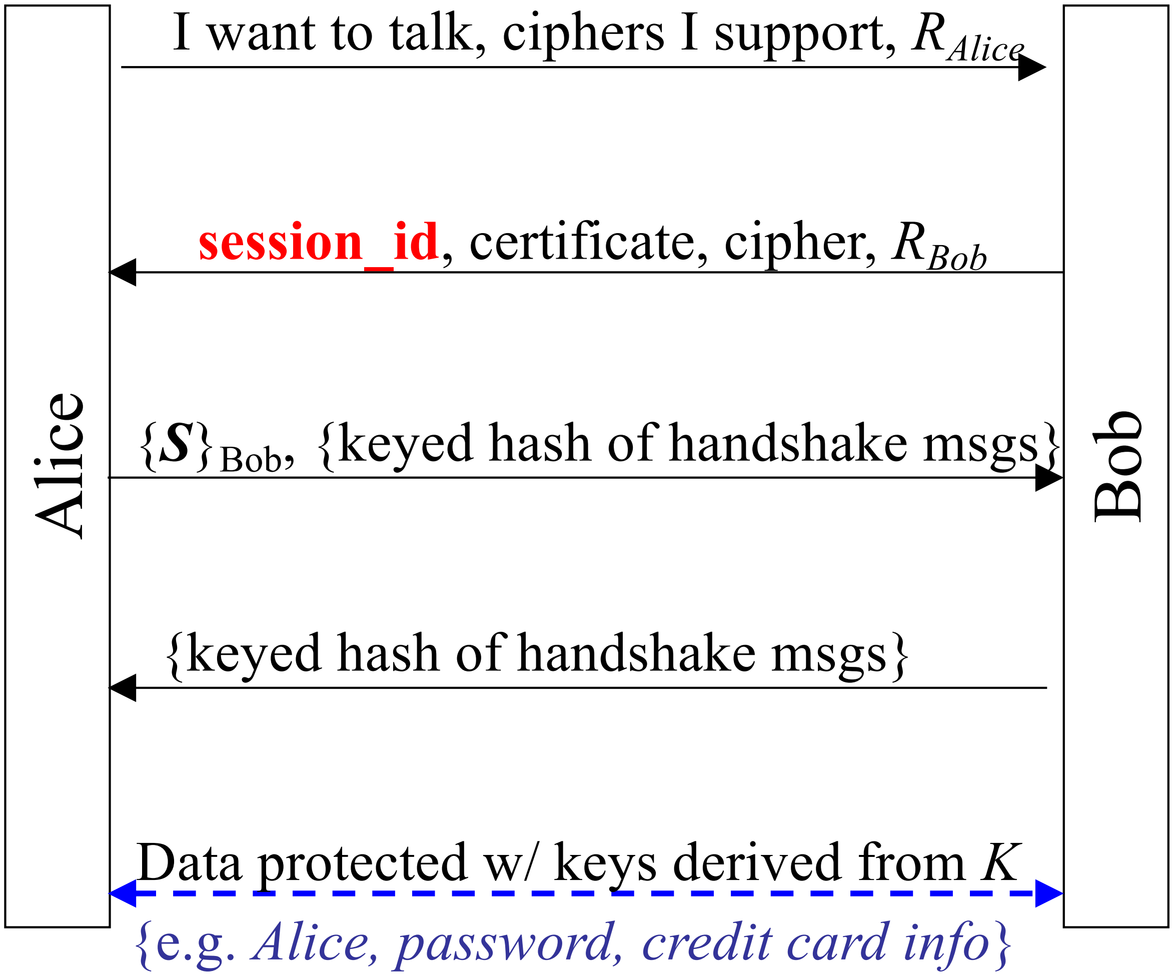
Data protected w/ keys derived from K

{e.g. *Alice, password, credit card info*}

One Session w/ Multiple Connnections

- From a long SSL session, after one connection is set up, many other *connections* can further be derived
  - Alice (a browser) and Bob (a web site) can have many connections, for instance
- Simplify the SSL for later connections between Alice and Bob
  - They have gone through the pain anyway . . .

Session Initiation



I want to talk, ciphers I support, R_{Alice}

session_id, certificate, cipher, R_{Bob}

$\{\boldsymbol{S}\}_{\mathrm{Bob}}$, {keyed hash of handshake msgs}

{keyed hash of handshake msgs}

Data protected w/ keys derived from K

{e.g. *Alice, password, credit card info*}

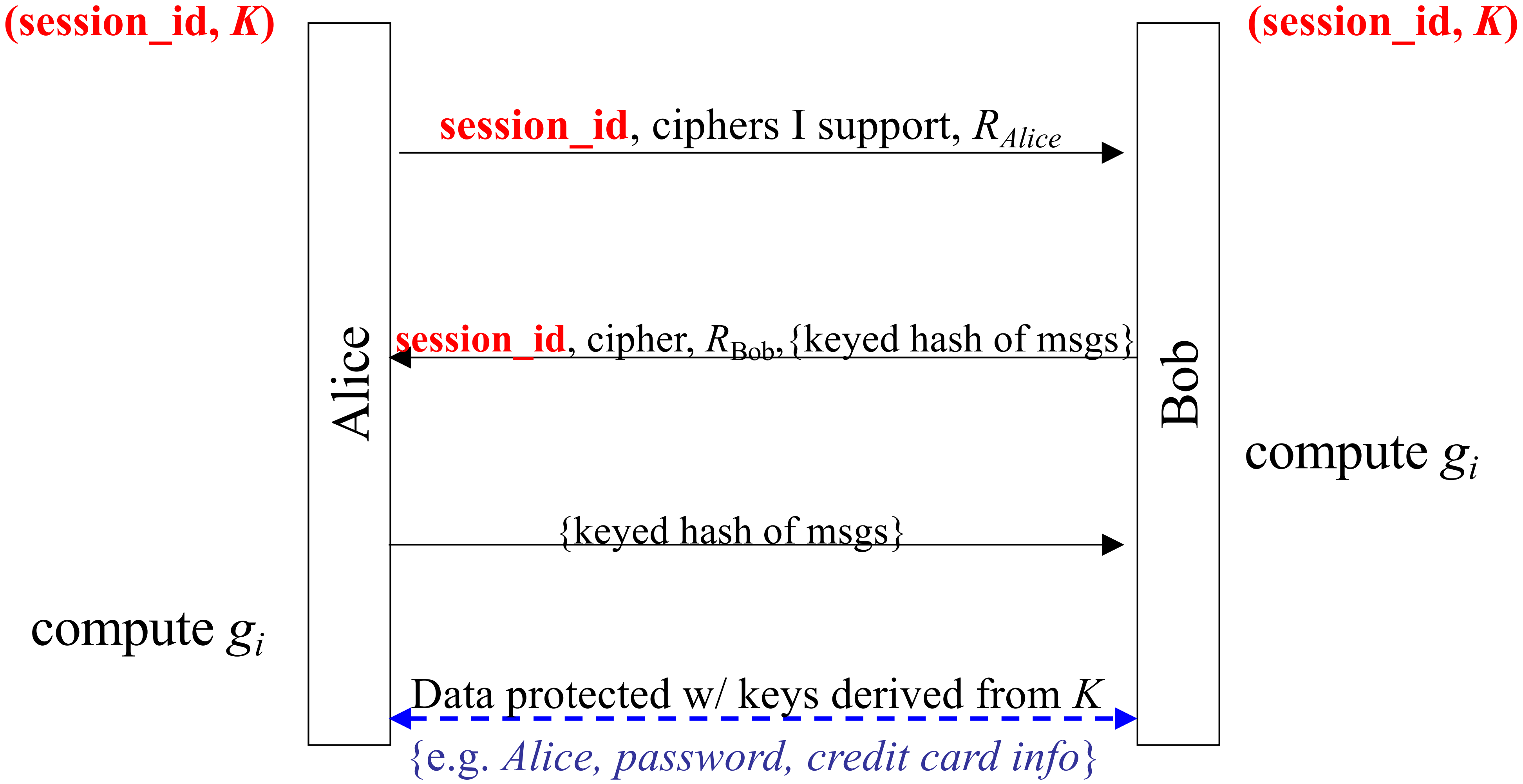
Choose secret $\boldsymbol{S}$, compute $K=f(\boldsymbol{S}, R_{Alice}, R_{bob})$

remembers (session_id, *K*)

compute $K=f(\boldsymbol{S}, R_{Alice}, R_{bob})$

remembers (session_id, *K*)

Session Resumption



(**session_id**, ***K***)

Alice

session_id, ciphers I support, R_{Alice}

session_id, cipher, R_{Bob}, {keyed hash of msgs}

compute g_i

{keyed hash of msgs}

compute g_i

Data protected w/ keys derived from K

{e.g. *Alice, password, credit card info*}

Bob

(**session_id**, ***K***)

SSL/TLS is Asymmetrical

- Alice authenticated Bob
- But Bob does not authenticate Alice
  - Until Alice login to Bob
  - Could be Mallory handshaking with Bob
- SSL/TLS can be enhanced for mutual authentication
  - If the client has a certificate

Firewalls and Intrusion Detection Systems

Learning Objectives

- Basic concepts of firewalls (functions, types, configurations)
- Intrusion detection systems (how each type works)

What is a Firewall

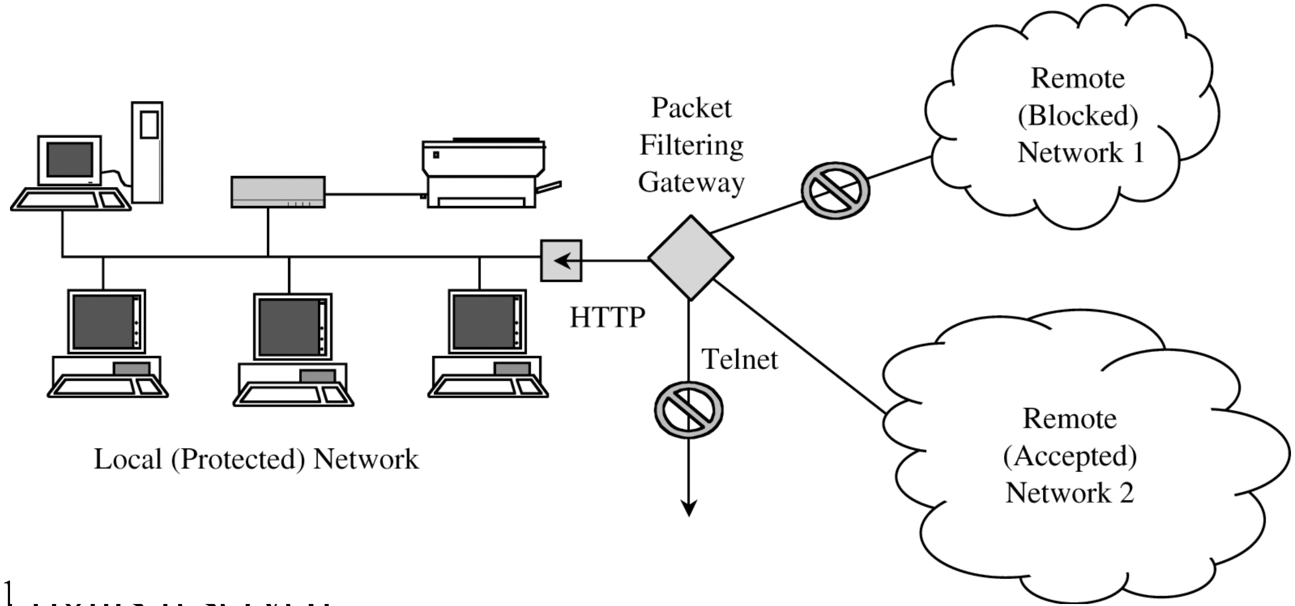
- A device that filters all traffic between a protected or “inside” network and a less trustworthy or “outside” network
- A special form of *reference monitor*
  - Default permit vs. default deny

Types of Firewalls

- Packet filtering
- Stateful inspection firewalls
- Application proxies
- Personal firewalls

Packet Filtering Firewall

- The simplest
  - Sometimes most effective
- On the basis of packet address (source or destination) or specific protocol type.

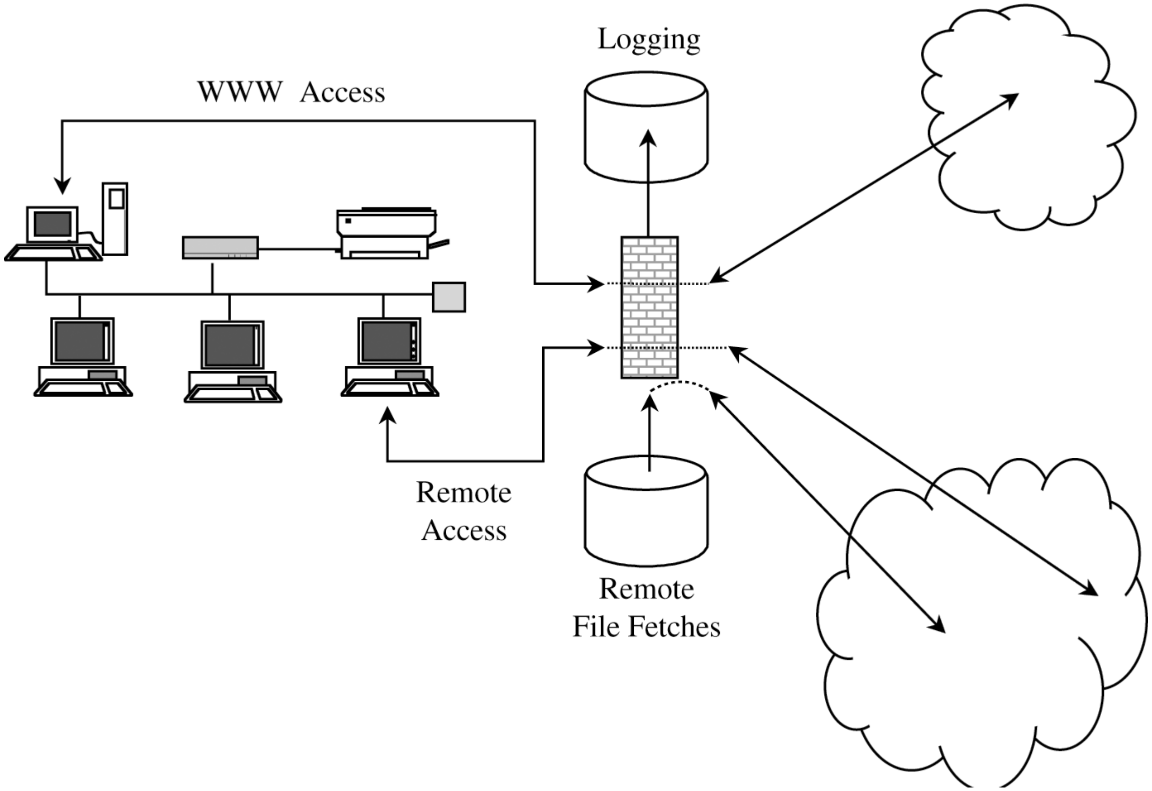


Stateful Inspection Firewall

- Maintains state information from one packet to another in the input stream
- Useful when an attacker breaks an attack into multiple packets
- The firewall can track the sequence of packets and conditions from one packet to another to thwart the attack

Application Proxy

- Inspect the application data



Personal Firewalls

- An application running on a workstation to block unwanted traffic from the network
- E.g., Combining virus scanner with the personal firewall
  - Forward all incoming packets to the virus scanner

Intrusion Detection Systems

- Signature-based vs. anomaly-based
- Host-based vs. network-based

- False negatives vs. false positives

Midterm

- February 17; Open book, open notes
- Will be available on Canvas on 2/17 10:00 AM
- You then work on a Microsoft Word document
  - We plan to provide a fillable PDF too
- Must submit by 2/17 11:30 AM
  - Give yourself some time to upload your midterm
  - You may submit it multiple times, so long as it's before it's due
  - Tip: maybe try a submission earlier, then continue to work on it until you submit the final version
- I will be available via zoom