

# **CSC4200/5200 – COMPUTER NETWORKING**

**Instructor: Susmit Shannigrahi**

**NETWORK SECURITY - CONTINUED**

**sshannigrahi@tnitech.edu**

**GTA: dereddick42@students.tnitech.edu**



# Security Roadmap

---

What is network security?

Principles of cryptography

*Message integrity*

Authentication

Securing TCP connections: SSL

Network layer security: IPsec

Operational security: firewalls and IDS

# What is network security?

 *confidentiality*: only sender, intended receiver should “understand” message contents

- sender encrypts message
- receiver decrypts message

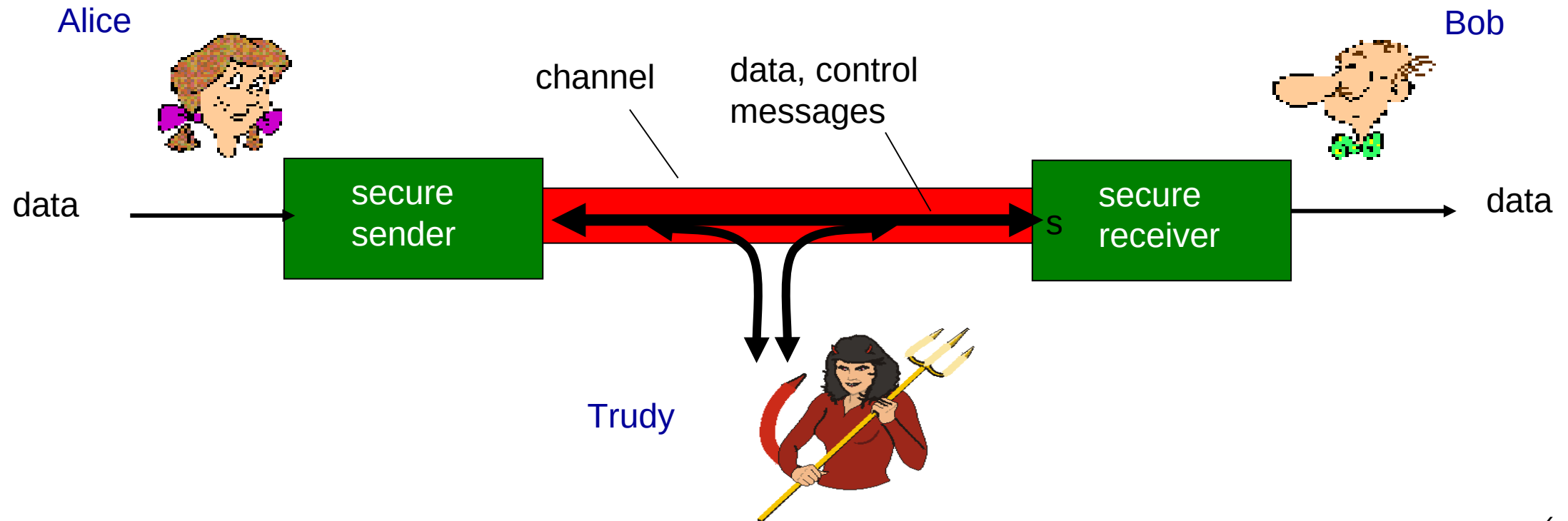
*authentication*: sender, receiver want to confirm identity of each other

*message integrity*: sender, receiver want to ensure message not altered (in transit, or afterwards) without detection

*access and availability*: services must be accessible and available to users

# Friends and enemies: Alice, Bob, Trudy

- Bob and Alice want to communicate “securely”
- Trudy may intercept, delete, add messages



# Some example problems

---

- **eavesdrop**: intercept messages
- actively **insert** messages into connection
- **impersonation**: can fake (spoof) source address in packet (or any field in packet)
- **hijacking**: “take over” ongoing connection by removing sender or receiver, inserting himself in place
- **denial of service**: prevent service from being used by others (e.g., by overloading resources)

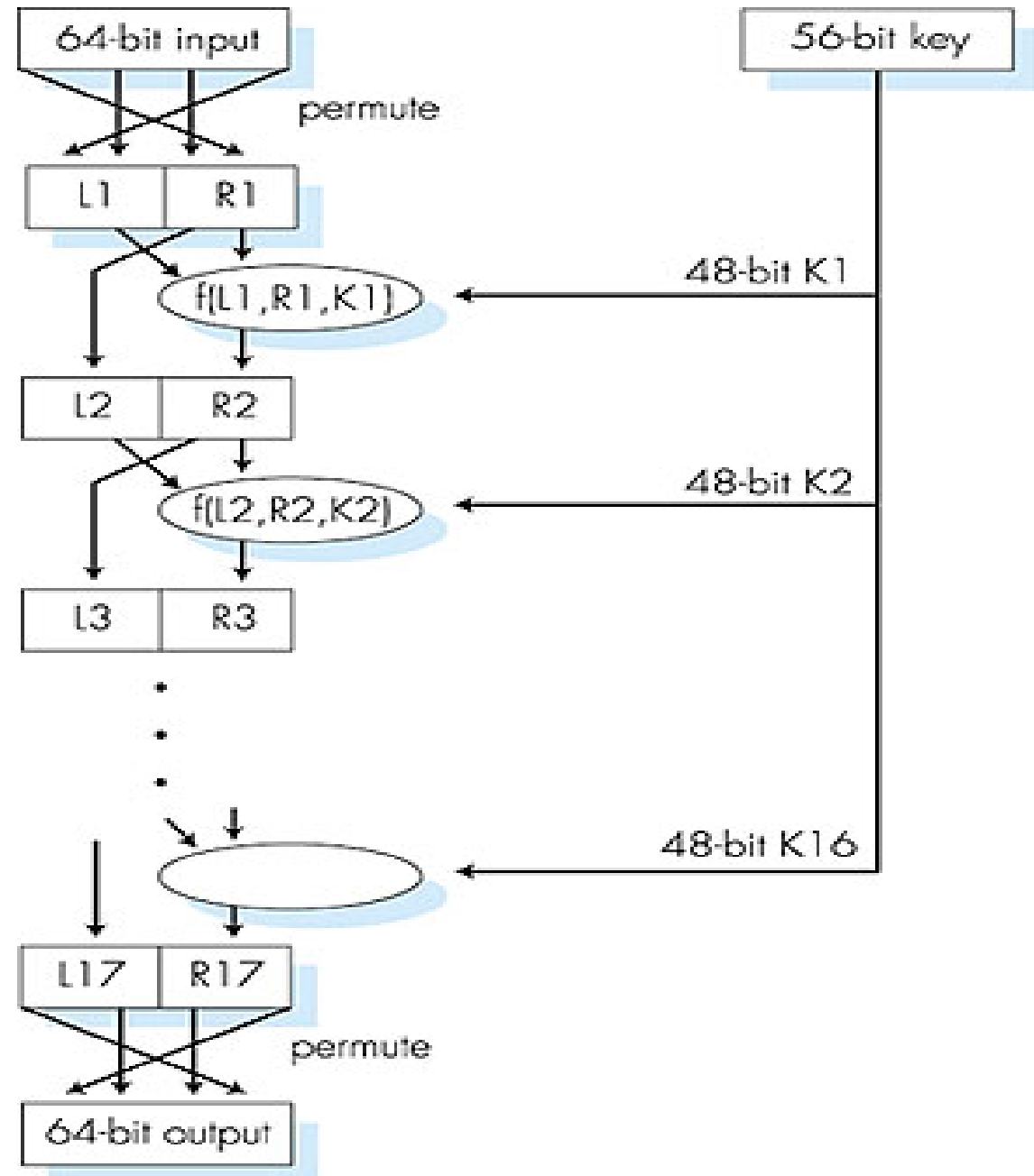
# Symmetric key crypto: DES

## *DES operation*

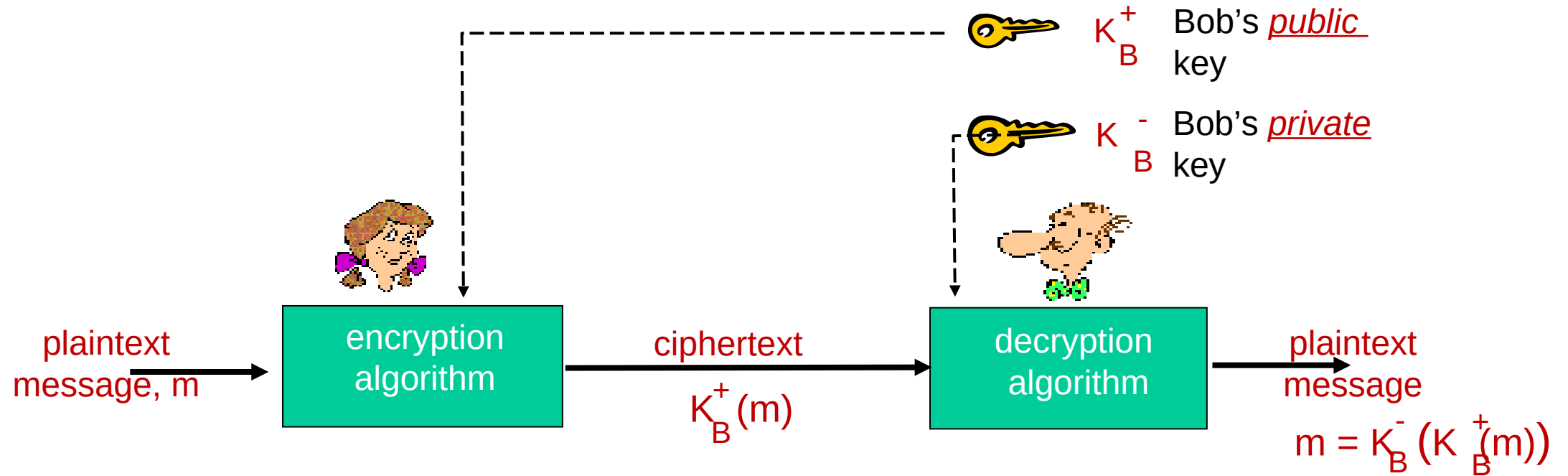
initial permutation

16 identical “rounds” of  
function application, each  
using different 48 bits of  
key

final permutation



# Public key cryptography



# Roadmap



What is network security?

Principles of cryptography

Message integrity

Authentication

*Securing TCP connections: SSL*

Network layer security: IPsec

Operational security: firewalls and IDS



# Authentication

*Goal:* Bob wants Alice to “prove” her identity to him

*Protocol ap1.0:* Alice says “ I am Alice”



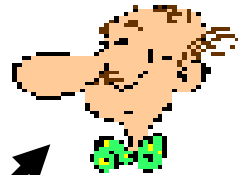
Failure scenario??



# Authentication

*Goal:* Bob wants Alice to “prove” her identity to him

*Protocol ap1.0:* Alice says “I am Alice”

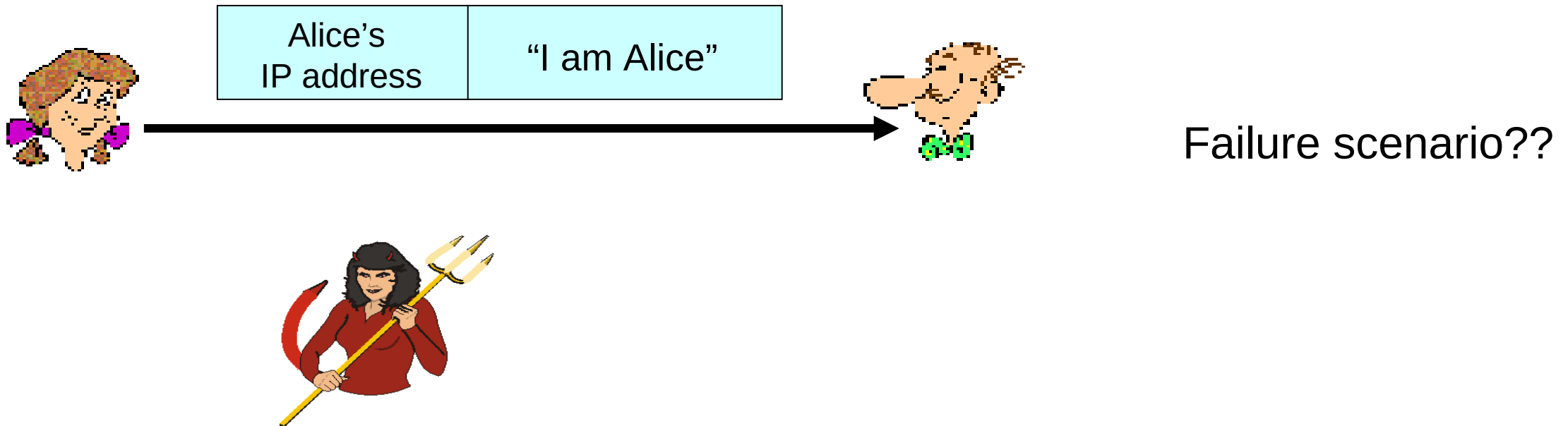


“I am Alice”

in a network,  
Bob can not “see” Alice, so Trudy  
simply declares  
herself to be Alice

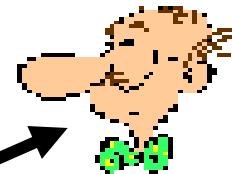
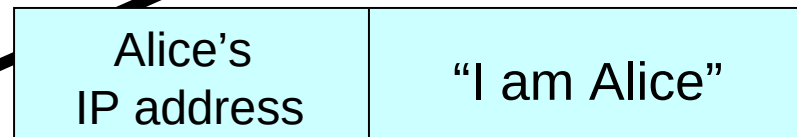
# Authentication: another try

*Protocol ap2.0:* Alice says “I am Alice” in an IP packet containing her source IP address



# Authentication: another try

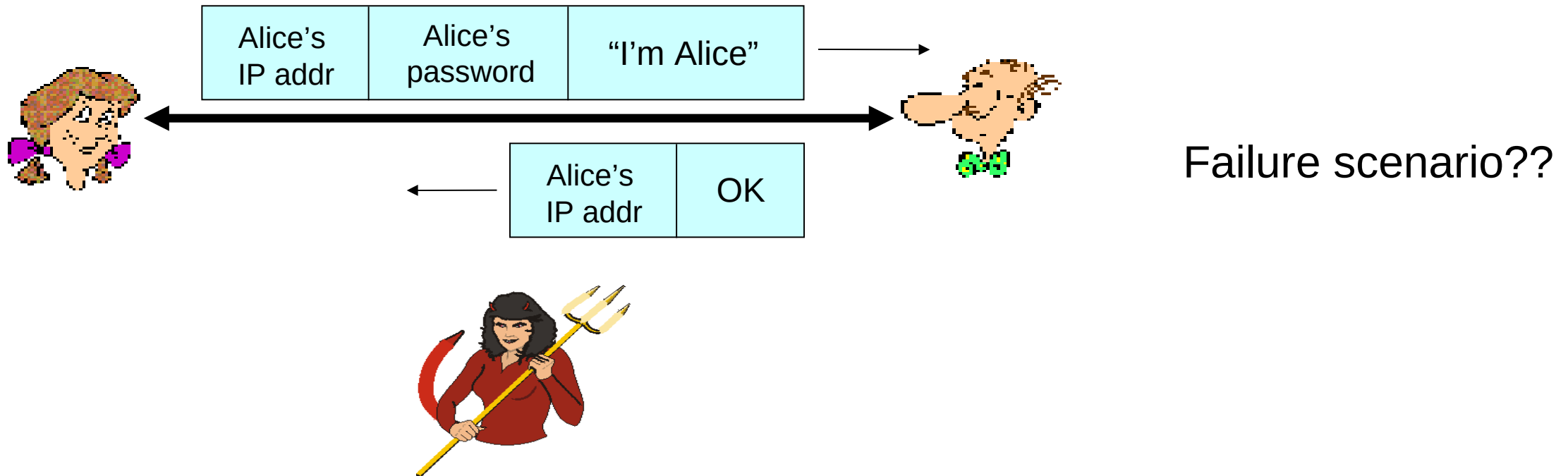
*Protocol ap2.0:* Alice says “I am Alice” in an IP packet containing her source IP address



Trudy can create a packet “spoofing” Alice’s address

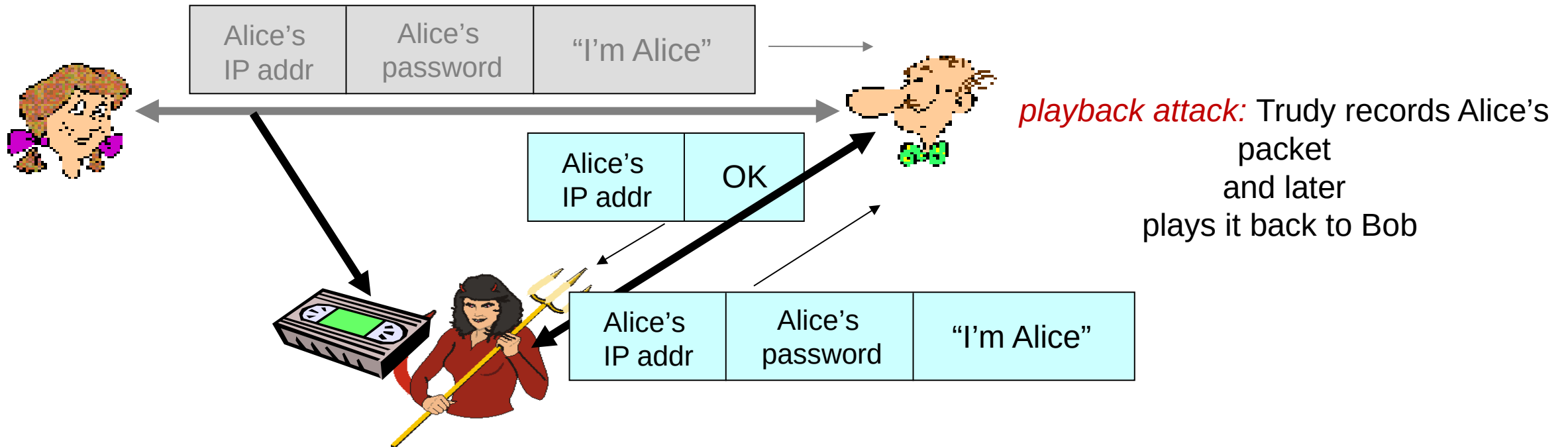
# Authentication: another try

*Protocol ap3.0:* Alice says “ I am Alice” and sends her secret password to “ prove” it.



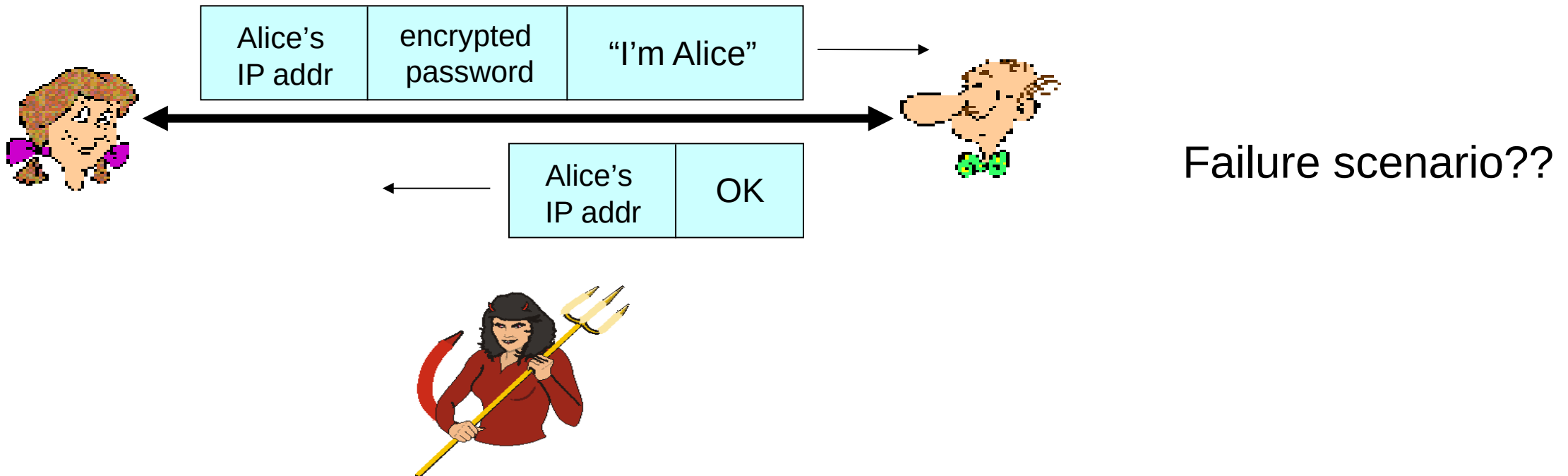
# Authentication: another try

*Protocol ap3.0:* Alice says “ I am Alice” and sends her secret password to “ prove” it.



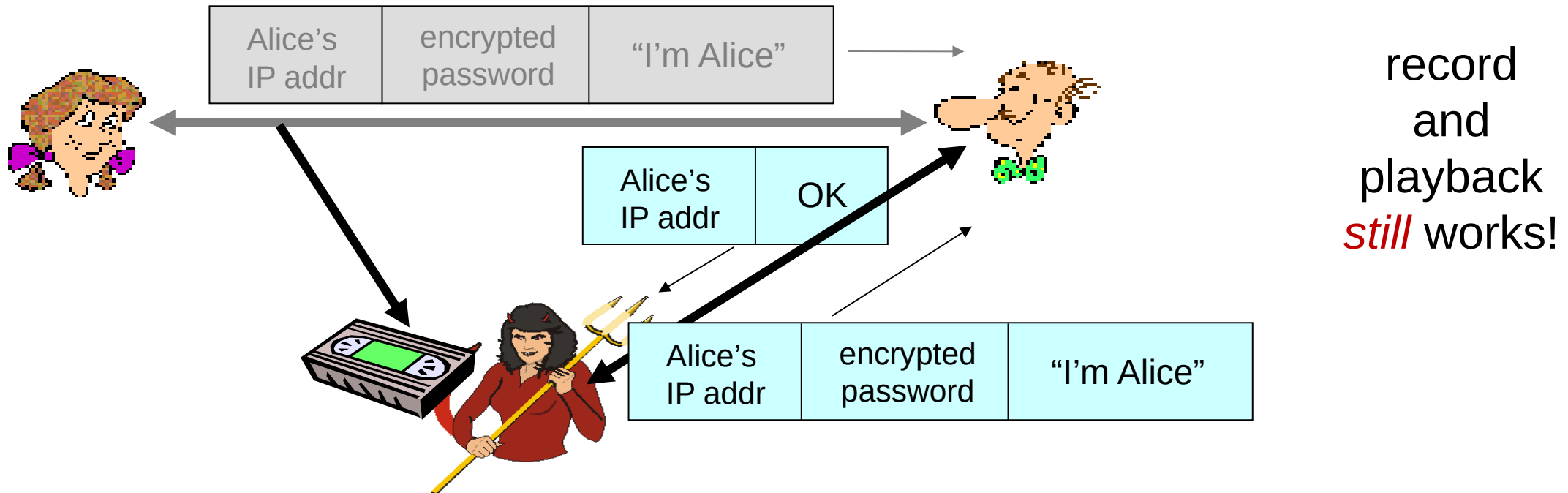
# Authentication: yet another try

*Protocol ap3.1:* Alice says “ I am Alice” and sends her *encrypted* secret password to “ prove” it.



# Authentication: yet another try

*Protocol ap3.1:* Alice says “ I am Alice” and sends her *encrypted* secret password to “ prove” it.



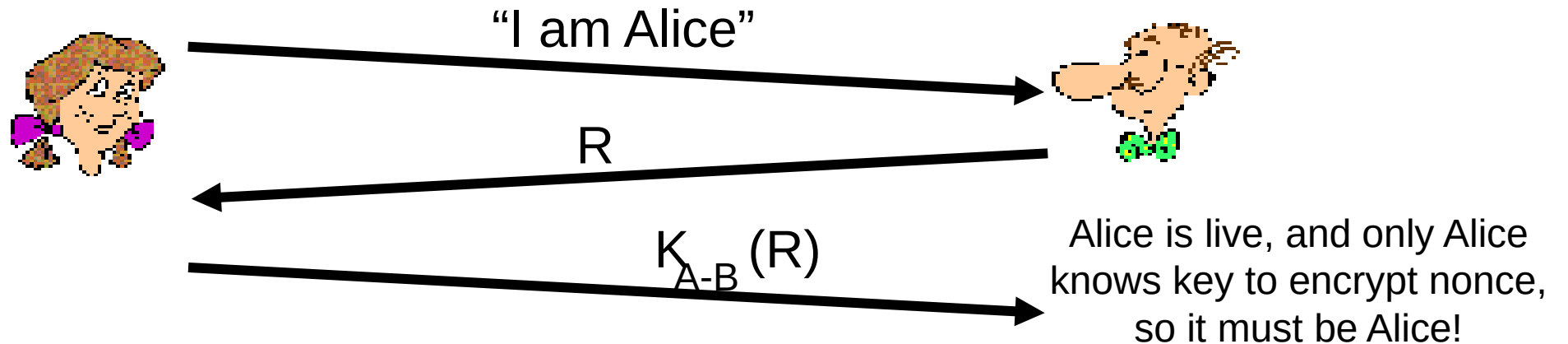


# Authentication: yet another try

**Goal:** avoid playback attack

**nonce:** number (R) used only *once-in-a-lifetime*

**ap4.0:** to prove Alice “live”, Bob sends Alice **nonce**, R. Alice must return R, encrypted with shared secret key



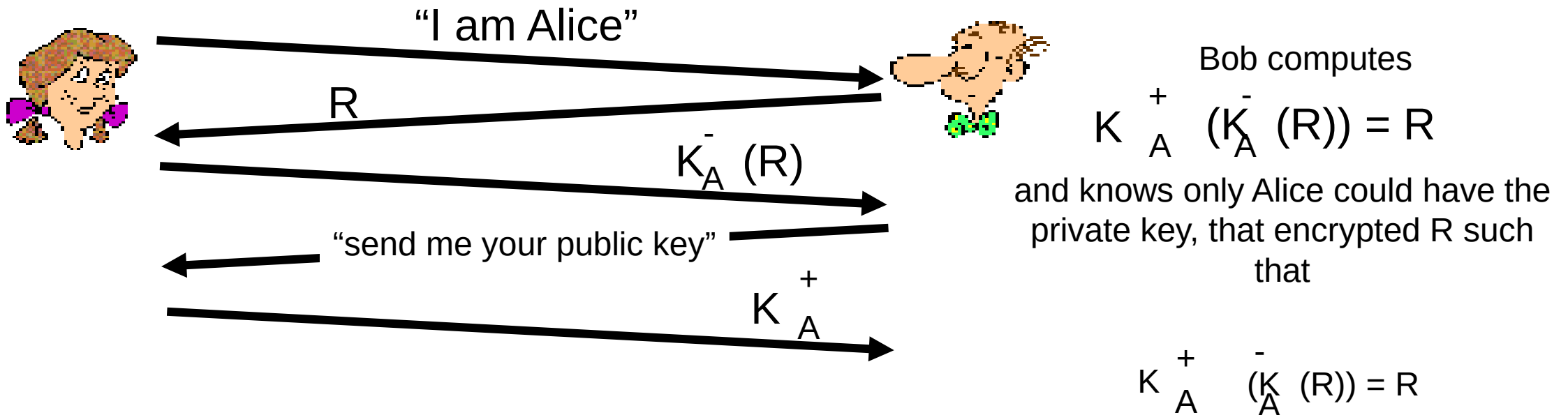
Failures, drawbacks?

# Authentication: ap5.0

ap4.0 requires shared symmetric key

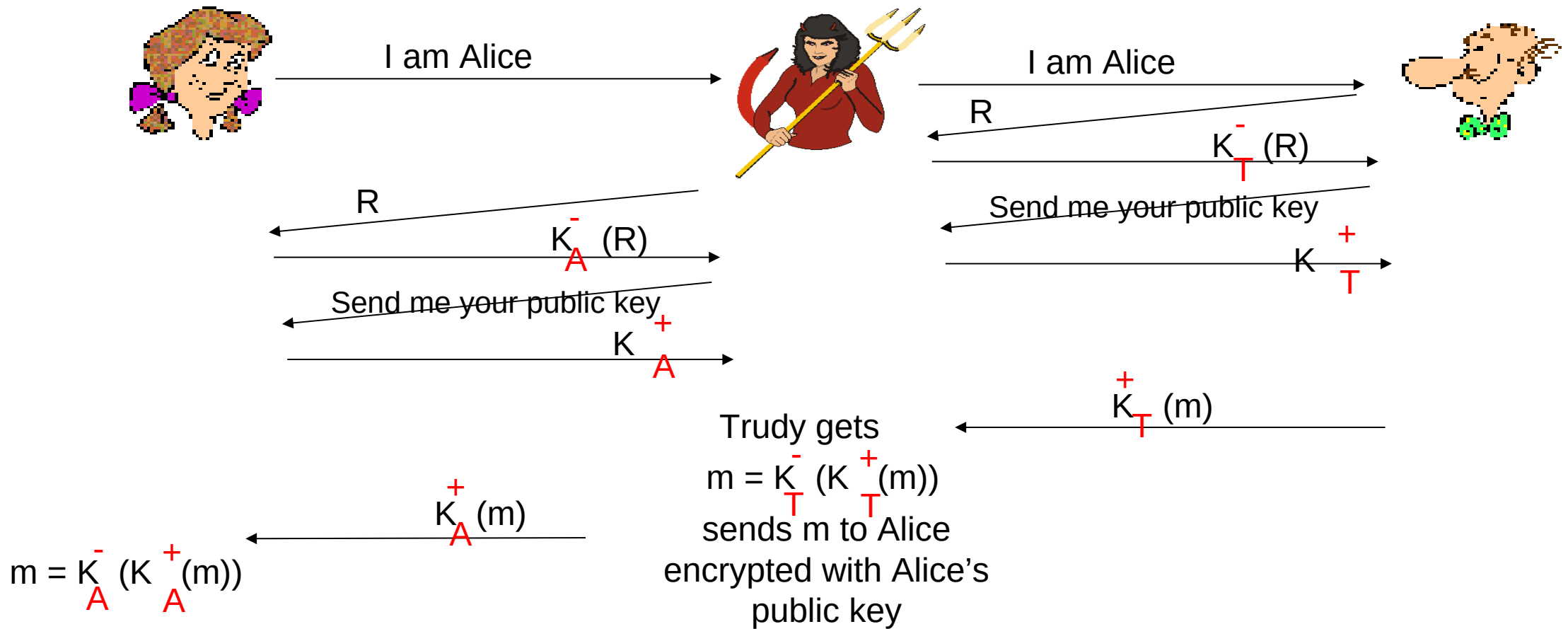
- can we authenticate using public key techniques?

*ap5.0*: use nonce, public key cryptography



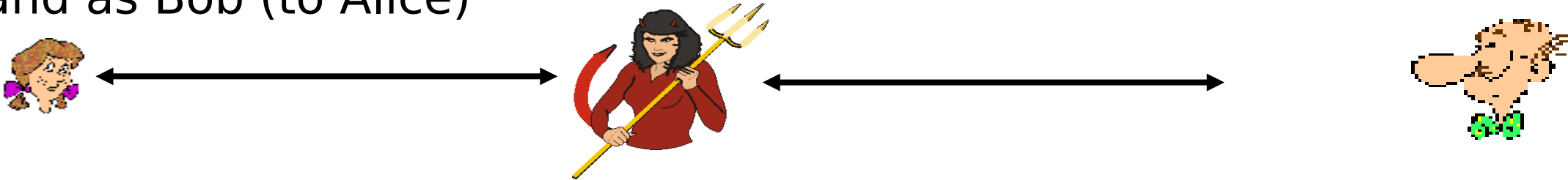
# ap5.0: security hole

*man (or woman) in the middle attack:* Trudy poses as Alice (to Bob) and as Bob (to Alice)



# ap5.0: security hole

*man (or woman) in the middle attack:* Trudy poses as Alice (to Bob) and as Bob (to Alice)



difficult to detect:

- ❖ Bob receives everything that Alice sends, and vice versa.
- ❖ problem is that Trudy receives all messages as well!

# Roadmap



What is network security?

Principles of cryptography

Message integrity

Authentication

*Securing TCP connections: SSL*

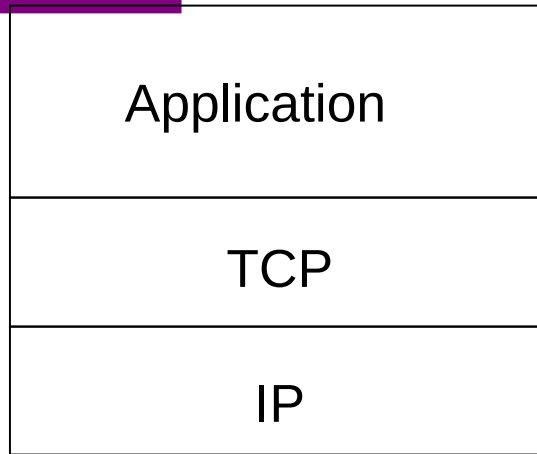
Network layer security: IPsec

Operational security: firewalls and IDS

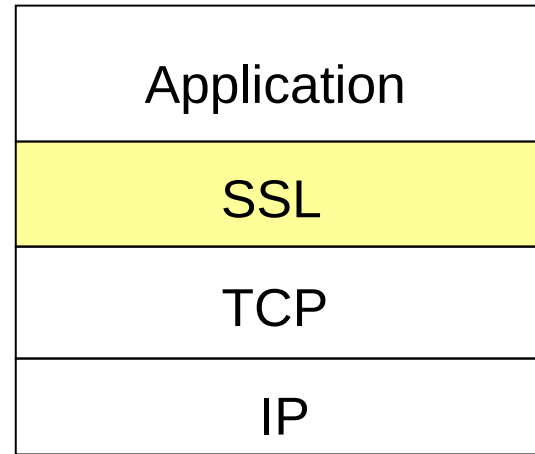
# SSL: Secure Sockets Layer

- ❖ widely deployed security protocol
  - supported by almost all browsers, web servers
  - https
  - billions \$/year over SSL
- ❖ mechanisms: [Woo 1994], implementation: Netscape
- ❖ variation -TLS: transport layer security, RFC 2246
- ❖ provides
  - *confidentiality*
  - *integrity*
  - *authentication*
- ❖ original goals:
  - Web e-commerce transactions
  - encryption (especially credit-card numbers)
  - Web-server authentication
  - optional client authentication
  - minimum hassle in doing business with new merchant
- ❖ available to all TCP applications
  - secure socket interface

# SSL and TCP/IP



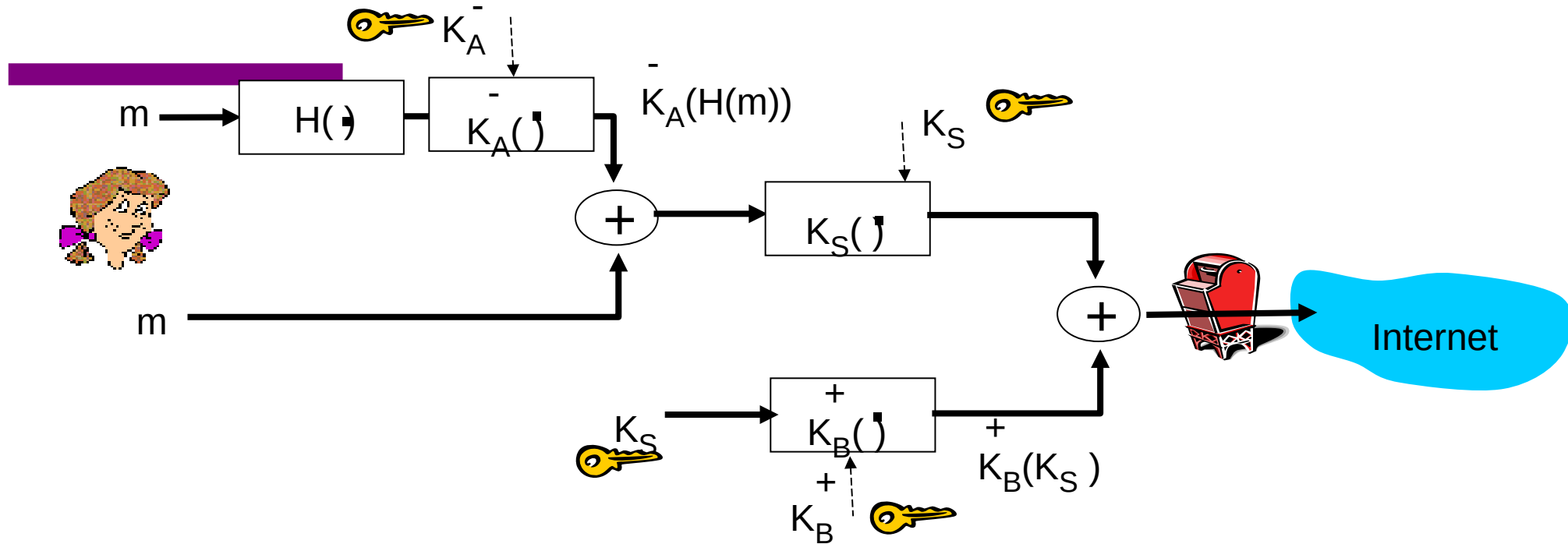
*normal application*



*application with SSL*

- ❖ SSL provides application programming interface (API) to applications
- ❖ C and Java SSL libraries/classes readily available

# Could do something like PGP:



- ❖ but want to send byte streams & interactive data
- ❖ want set of secret keys for entire connection
- ❖ want certificate exchange as part of protocol: handshake phase



# Real SSL: handshake (1)

## *Purpose*

1. server authentication
2. negotiation: agree on crypto algorithms
3. establish keys
4. client authentication (optional)

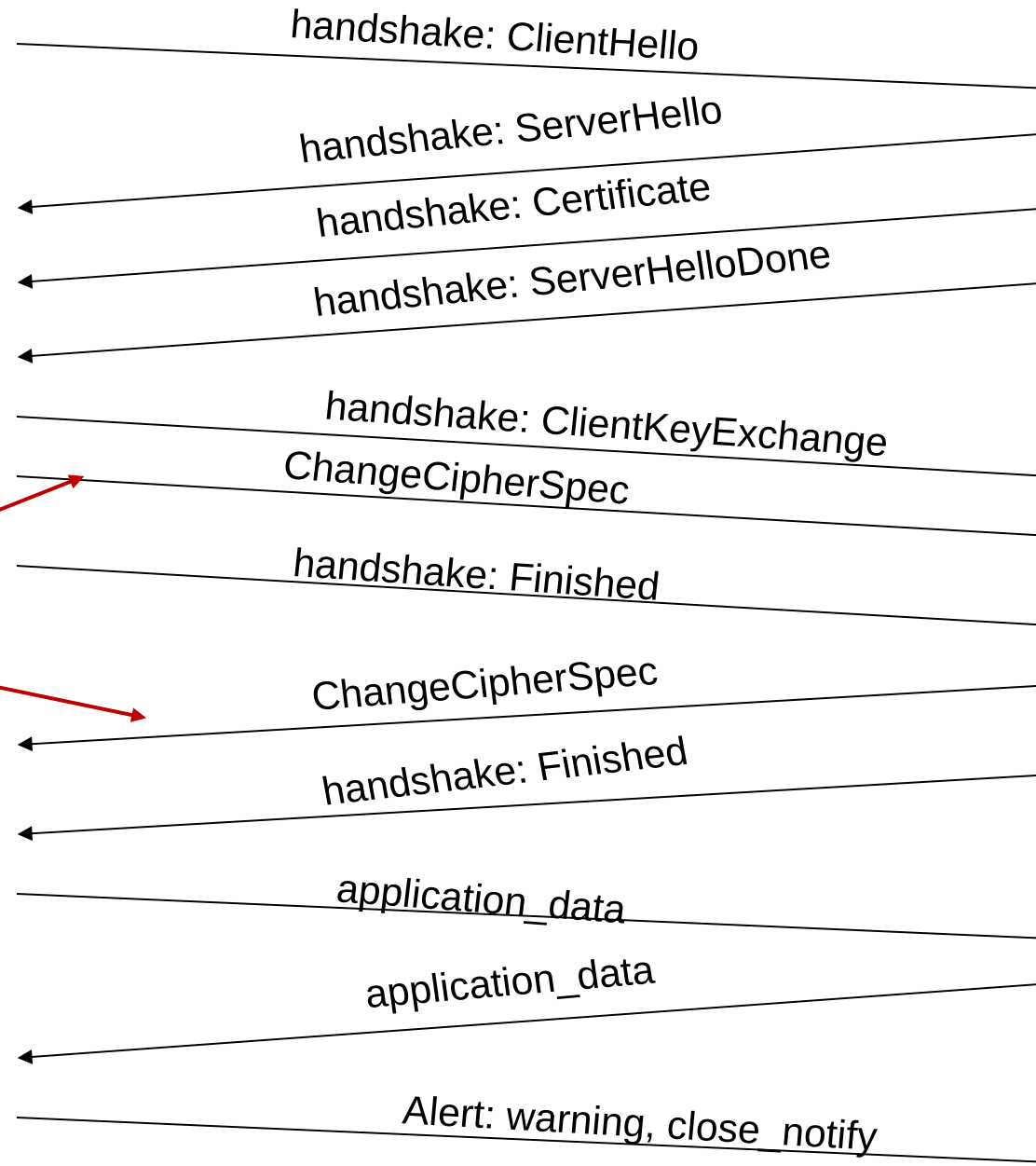
# Real SSL: handshake (2)

1. client sends list of algorithms it supports, along with client nonce
2. server chooses algorithms from list; sends back: choice + certificate + server nonce
3. client verifies certificate, extracts server's public key, generates pre\_master\_secret, encrypts with server's public key, sends to server
4. client and server independently compute encryption and MAC keys from pre\_master\_secret and nonces
5. client sends a MAC of all the handshake messages
6. server sends a MAC of all the handshake messages

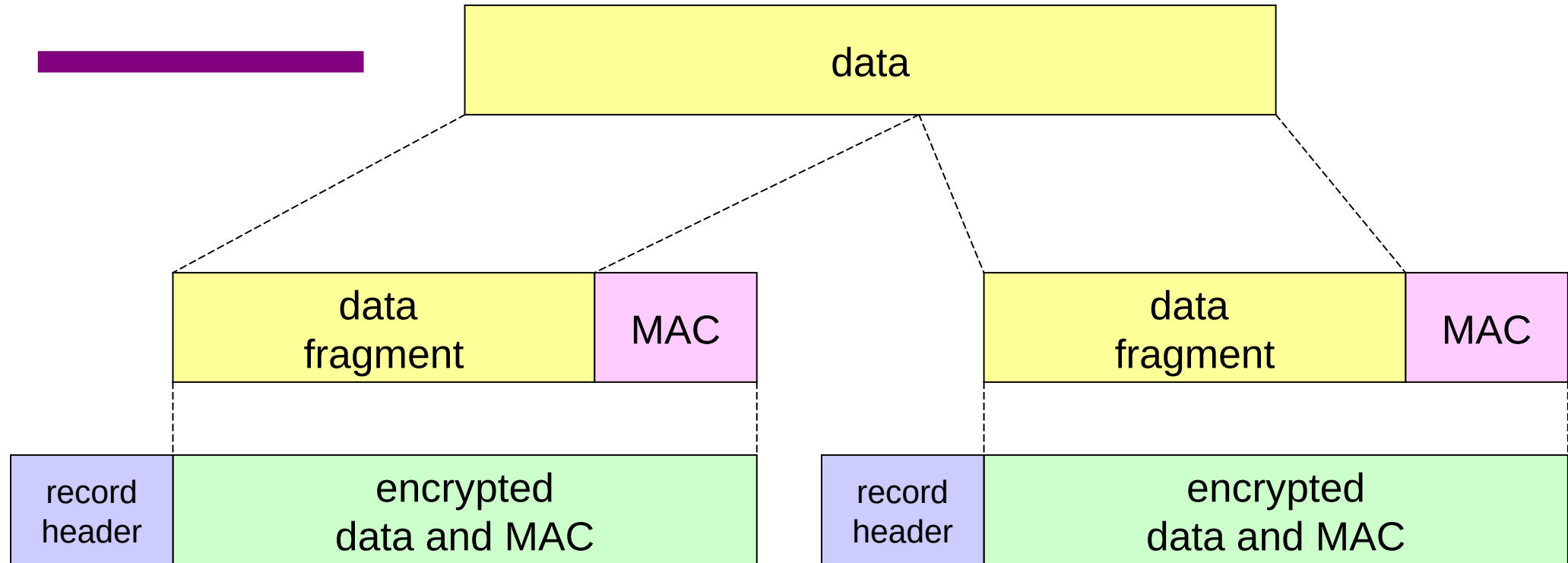
# Real SSL connection

*everything  
henceforth  
is encrypted*

TCP FIN follows



# SSL record protocol

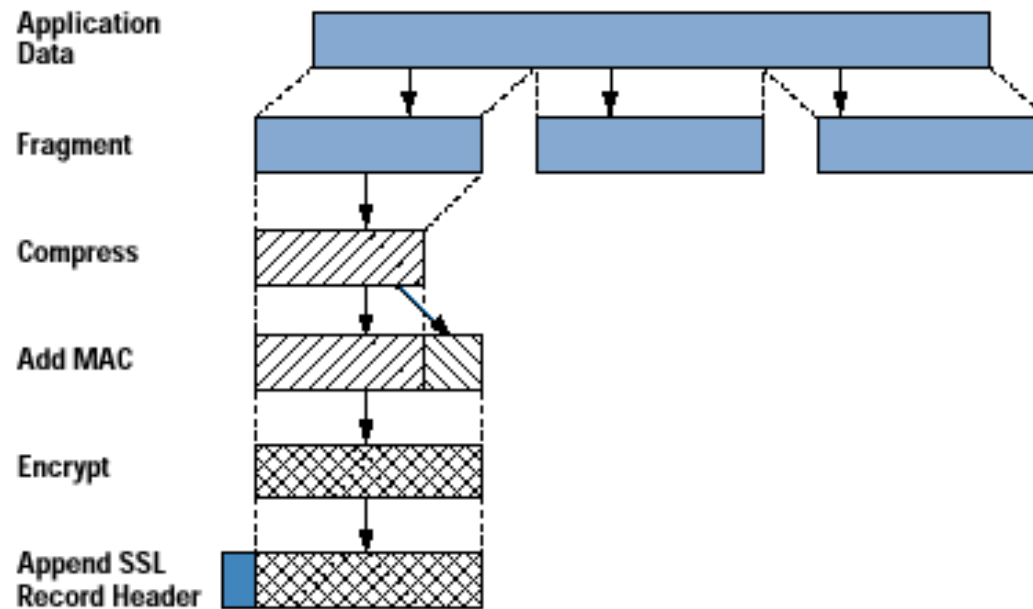


*record header:* content type; version; length

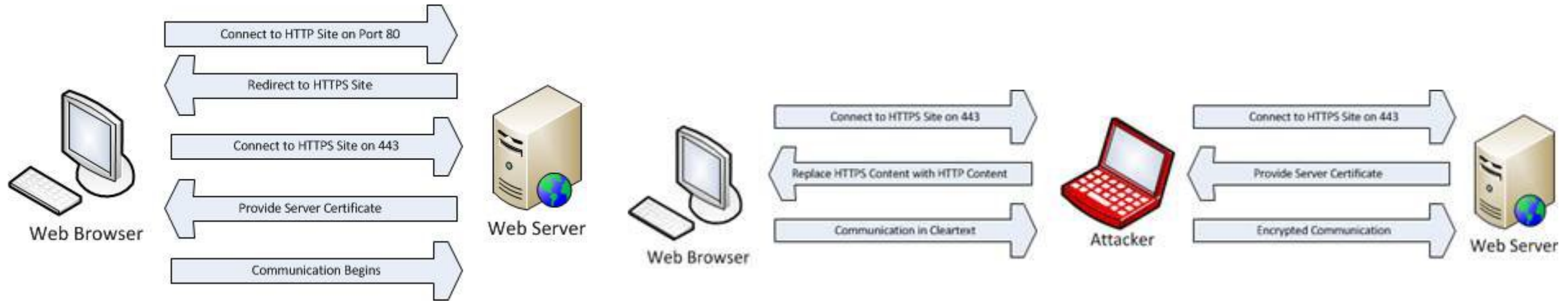
*MAC:* includes sequence number, MAC key  $M_x$

*fragment:* each SSL fragment  $2^{14}$  bytes (~16 Kbytes)

# Application and SSL



# SSL and HTTPS



# Roadmap

---

What is network security?

Principles of cryptography

Message integrity

Authentication

*Securing TCP connections: SSL*

Network layer security: VPN and IPsec

Operational security: firewalls and IDS

# What is network-layer confidentiality ?

*between two network entities:*

- sending entity encrypts datagram payload, payload could be:
  - TCP or UDP segment, ICMP message, OSPF message ....
- all data sent from one entity to other would be hidden:
  - web pages, e-mail, P2P file transfers, TCP SYN packets ...
- “blanket coverage”

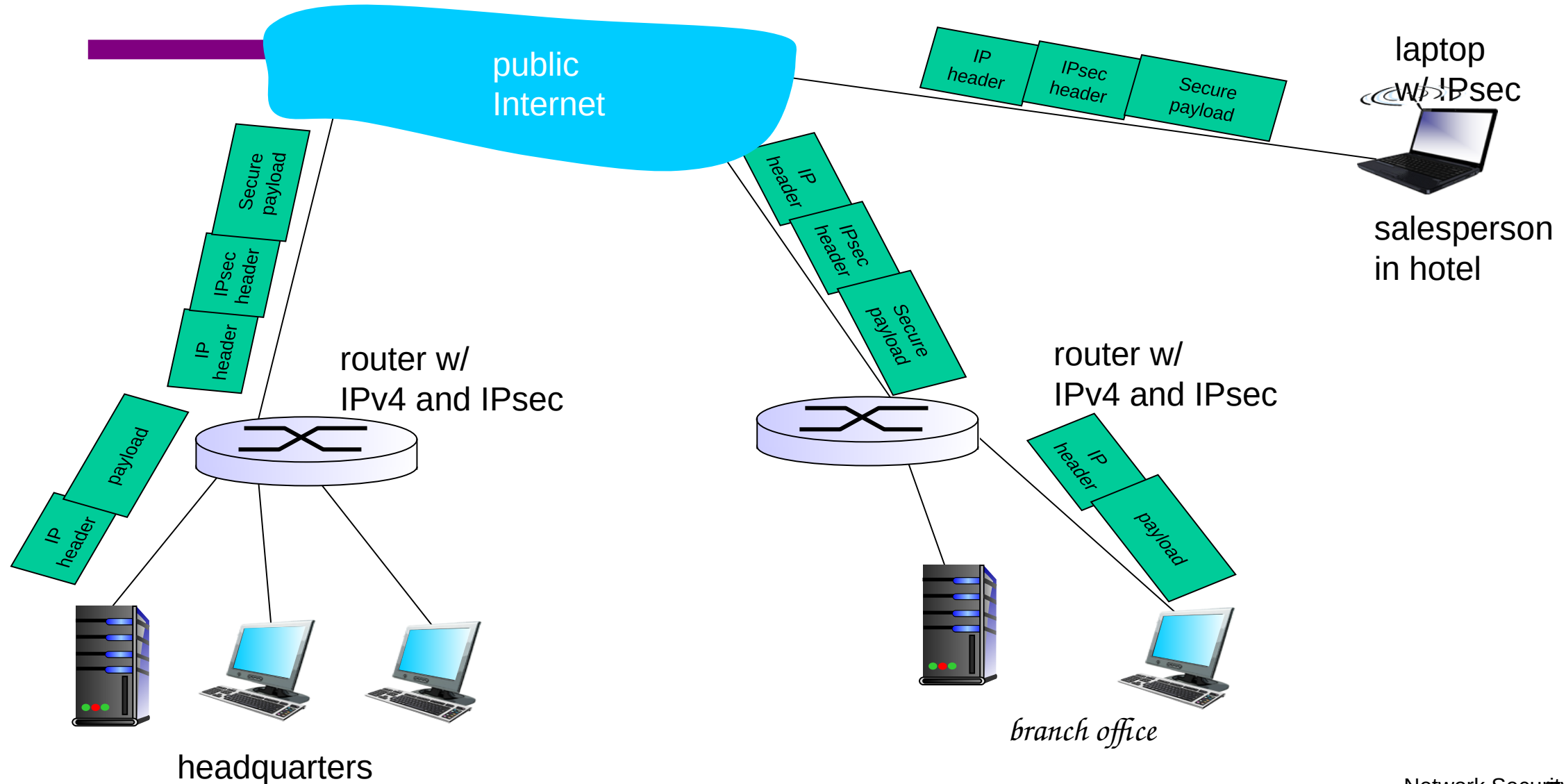


# Virtual Private Networks (VPNs)

## *motivation:*

- ❖ institutions often want private networks for security.
  - costly: separate routers, links, DNS infrastructure.
- ❖ VPN: institution's inter-office traffic is sent over public Internet instead
  - encrypted before entering public Internet
  - logically separate from other traffic

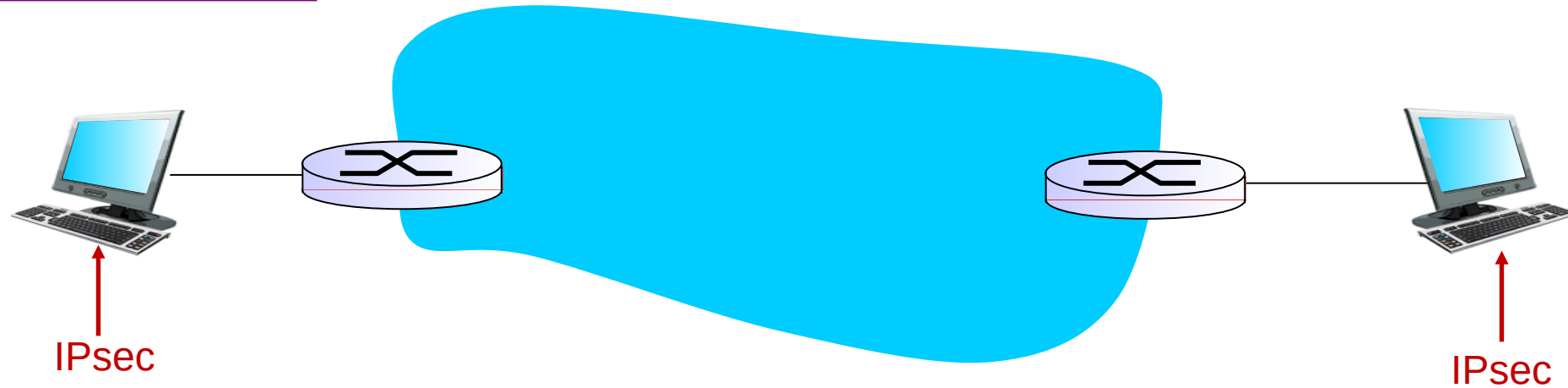
# Virtual Private Networks (VPNs)



# IPsec services

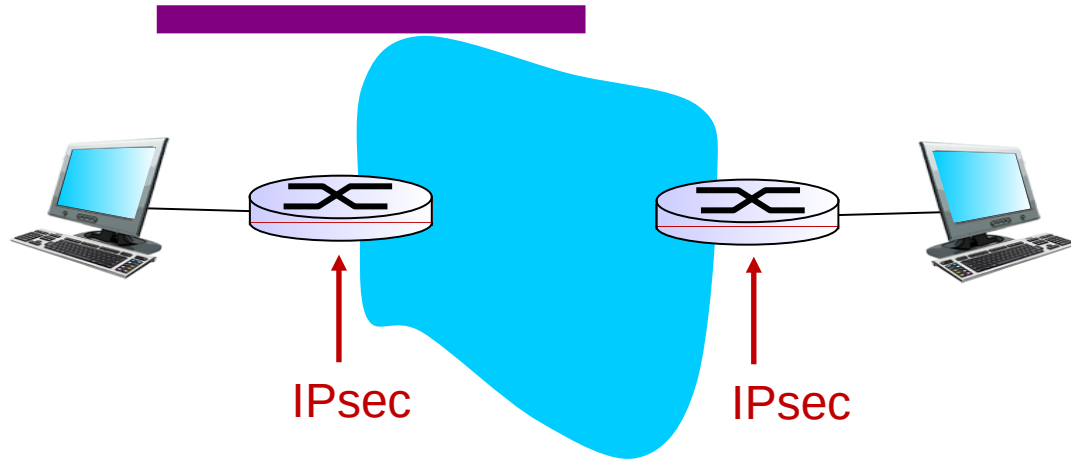
- data integrity
  - origin authentication
  - replay attack prevention
  - confidentiality
- 
- two protocols providing different service models:
    - AH
    - ESP

# IPsec transport mode

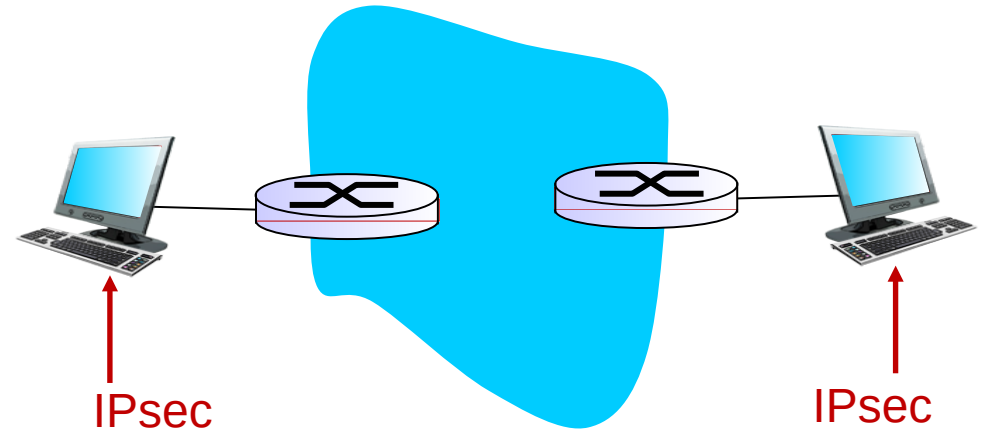


- IPsec datagram emitted and received by end-system
- protects upper level protocols

# IPsec – tunneling mode



- edge routers IPsec-aware




- ❖ hosts IPsec-aware

# Two IPsec protocols

- Authentication Header (AH) protocol
  - provides source authentication & data integrity but *not* confidentiality
- Encapsulation Security Protocol (ESP)
  - provides source authentication, data integrity, *and* confidentiality
  - more widely used than AH

# Four combinations are possible!



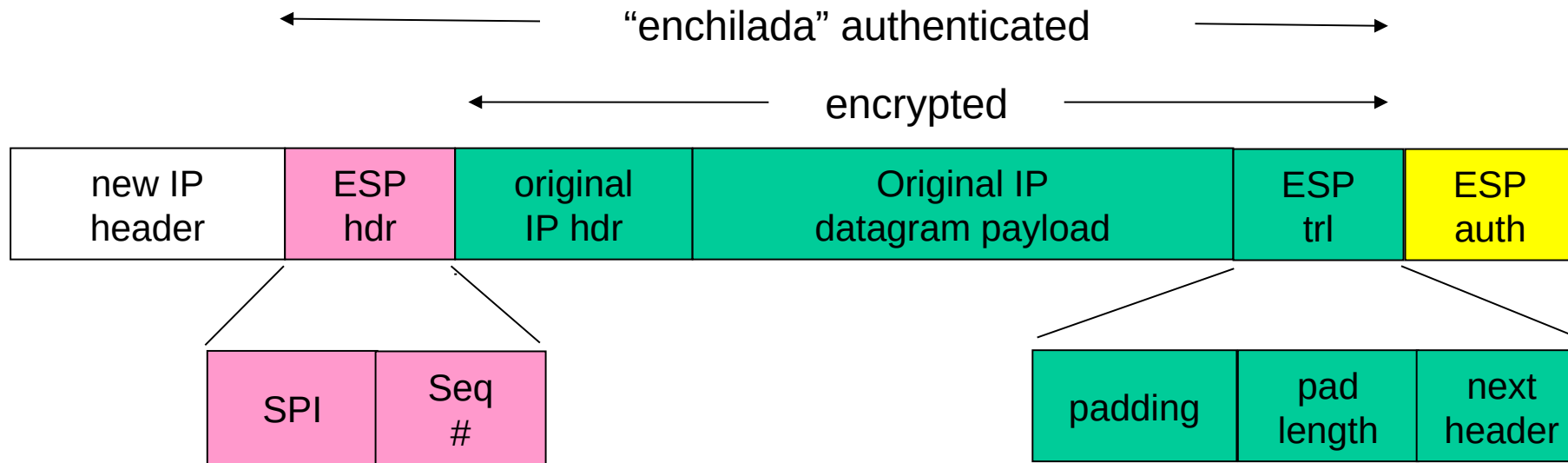
Host mode with AH	Host mode with ESP
Tunnel mode with AH	Tunnel mode with ESP



most common and  
most important

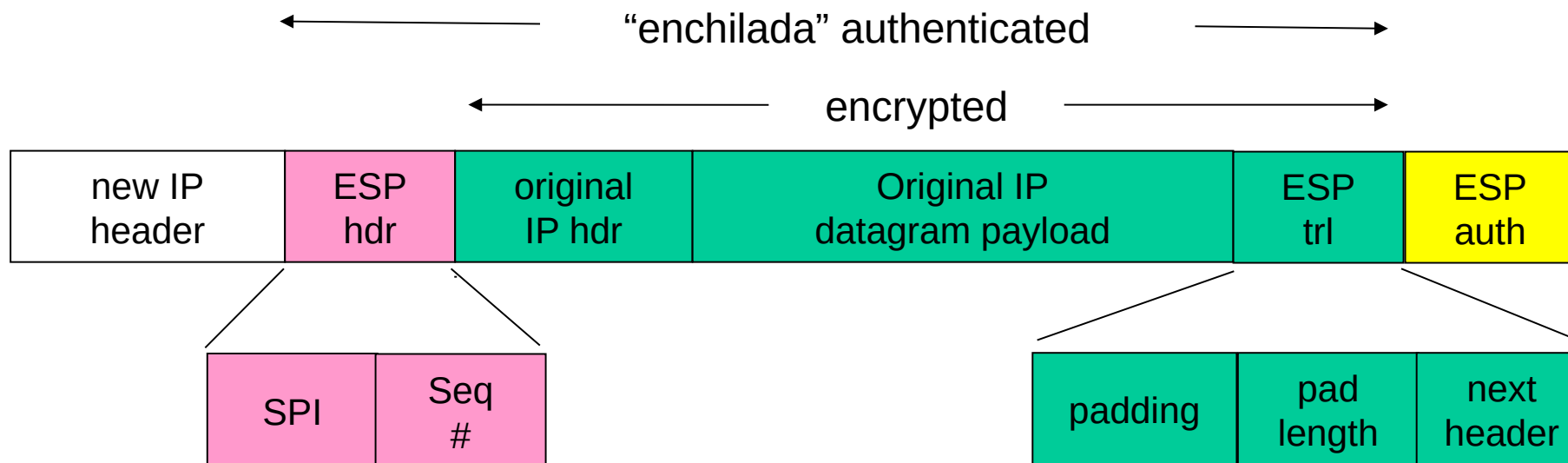
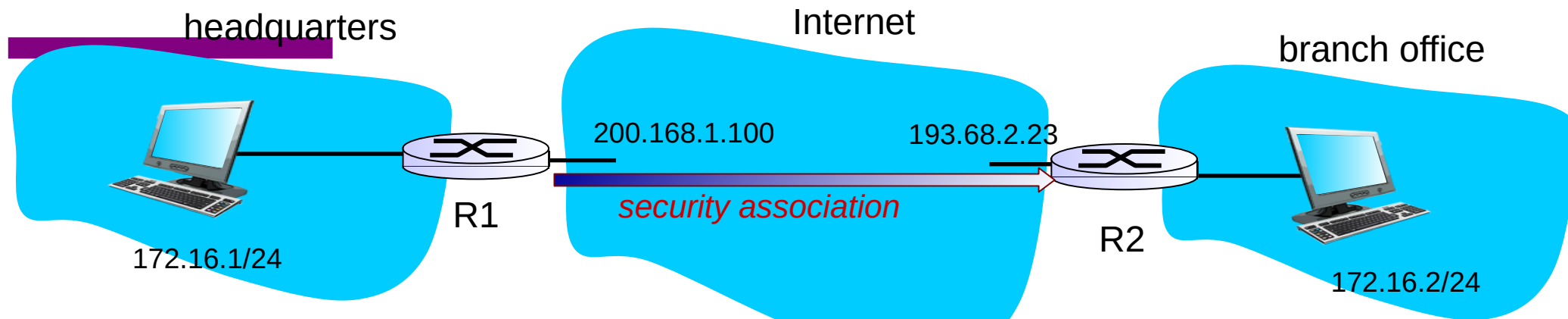
# IPsec datagram

focus for now on tunnel mode with ESP





# What happens?



# Roadmap

---

What is network security?

Principles of cryptography

Message integrity

Authentication

*Securing TCP connections: SSL*

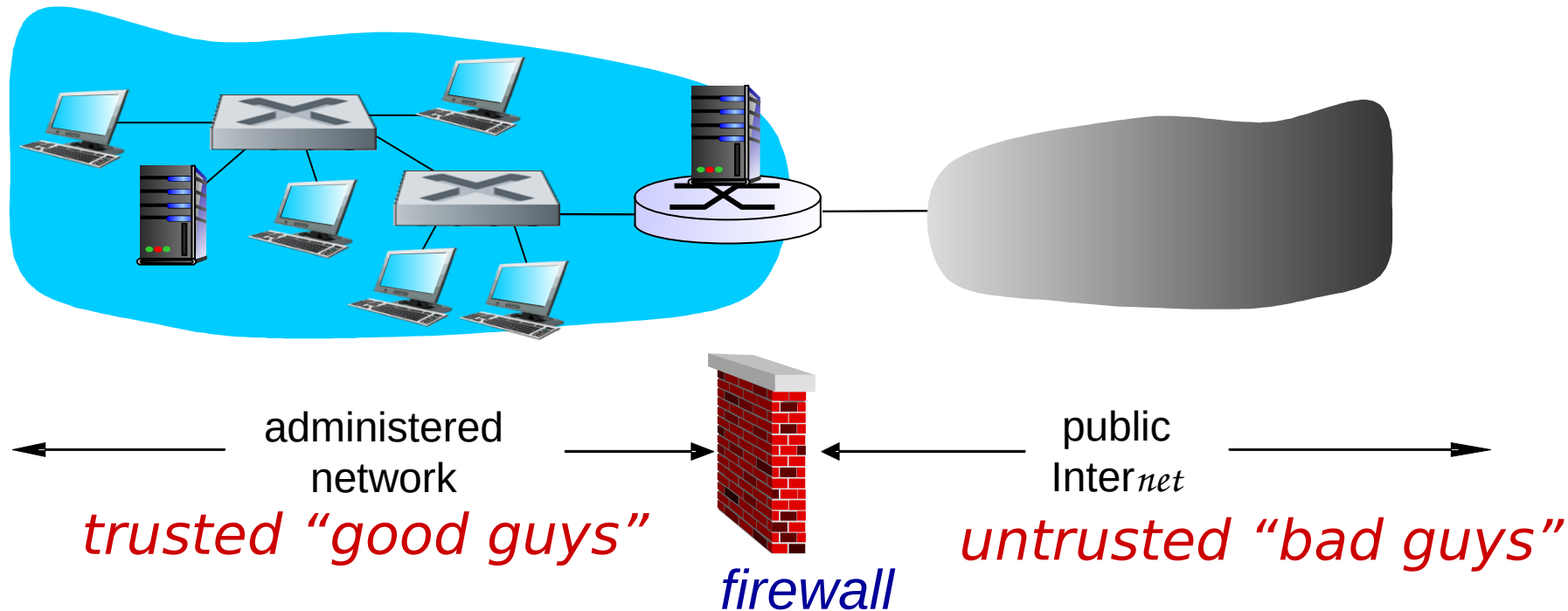
Network layer security: VPN and IPsec

Operational security: firewalls and IDS

# Firewalls

## *firewall*

isolates organization' s internal net from larger Internet, allowing some packets to pass, blocking others



# Firewalls: why

prevent denial of service attacks:

- ❖ SYN flooding: attacker establishes many bogus TCP connections, no resources left for “ real” connections

prevent illegal modification/access of internal data

- ❖ e.g., attacker replaces CIA’ s homepage with something else

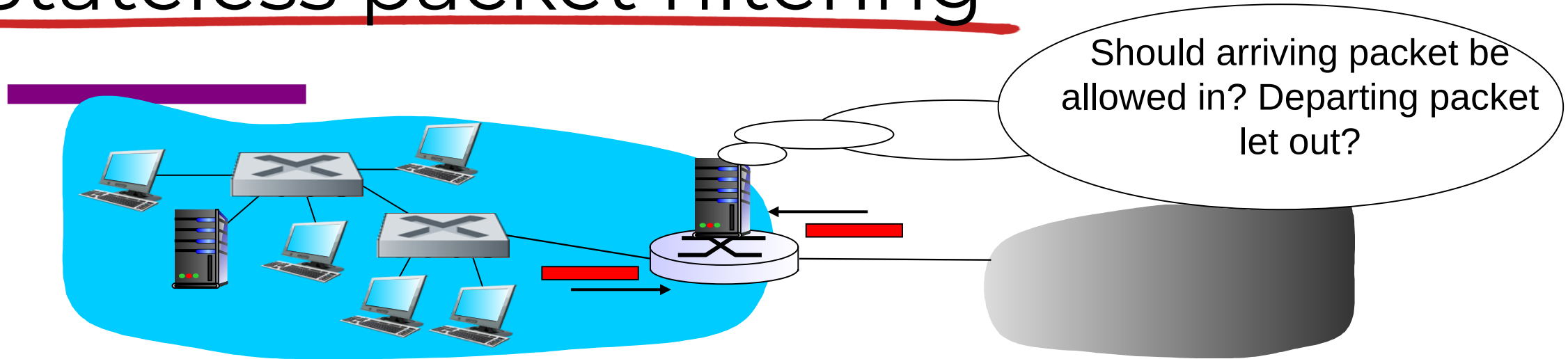
allow only authorized access to inside network

- ❖ set of authenticated users/hosts

three types of firewalls:

- ❖ stateless packet filters
- ❖ stateful packet filters
- ❖ application gateways

# Stateless packet filtering




- internal network connected to Internet via *router firewall*
- router *filters packet-by-packet*, decision to forward/drop packet based on:
  - source IP address, destination IP address
  - TCP/UDP source and destination port numbers
  - ICMP message type
  - TCP SYN and ACK bits

# Stateless packet filtering: example

- *example 1:* block incoming and outgoing datagrams with IP protocol field = 17 and with either source or dest port = 23
  - *result:* all incoming, outgoing UDP flows and telnet connections are blocked
- *example 2:* block inbound TCP segments with ACK=0.
  - *result:* prevents external clients from making TCP connections with internal clients, but allows internal clients to connect to outside.

# Stateless packet filtering: more examples

 <i>Policy</i>	<i>Firewall Setting</i>
No outside Web access.	Drop all outgoing packets to any IP address, port 80
No incoming TCP connections, except those for institution's public Web server only.	Drop all incoming TCP SYN packets to any IP except 130.207.244.203, port 80
Prevent Web-radios from eating up the available bandwidth.	Drop all incoming UDP packets - except DNS and router broadcasts.
Prevent your network from being used for a smurf DoS attack.	Drop all ICMP packets going to a "broadcast" address (e.g. 130.207.255.255).
Prevent your network from being tracerouted	Drop all outgoing ICMP TTL expired traffic

# Access Control Lists

- ❖ **ACL:** table of rules, applied top to bottom to incoming packets: (action, condition) pairs

action	source address	dest address	protocol	source port	dest port	flag bit
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	any
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	---
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	----
deny	all	all	all	all	all	all



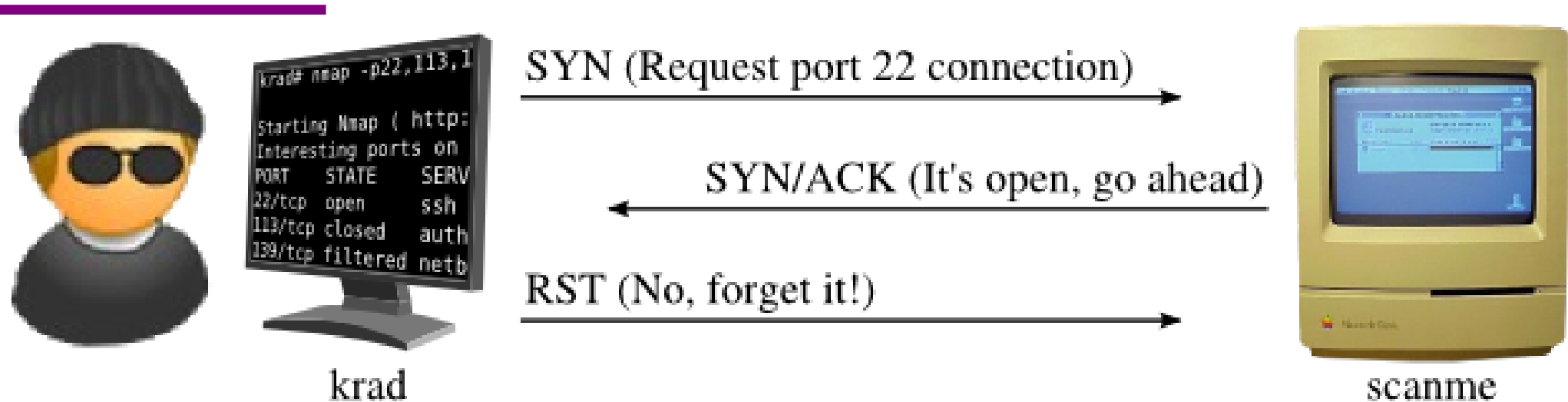
# Stateful packet filtering

- **stateless packet filter:** heavy handed tool
  - admits packets that “make no sense,” e.g., dest port = 80, ACK bit set, even though no TCP connection established:

action	source address	dest address	protocol	source port	dest port	flag bit
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK

- ❖ **stateful packet filter:** track status of every TCP connection
  - track connection setup (SYN), teardown (FIN): determine whether incoming, outgoing packets “ makes sense”
  - timeout inactive connections at firewall: no longer admit packets

# Stateless packet filtering: problems



If network security interests you: <https://nmap.org/book/>

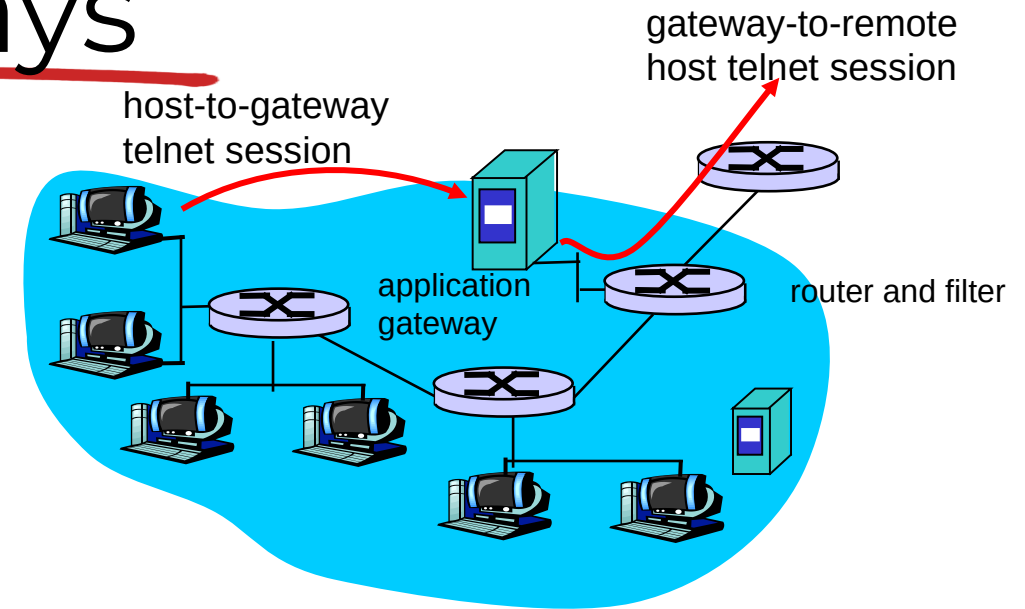
# Stateful packet filtering

- ❖ ACL augmented to indicate need to check connection state table before admitting packet

action	source address	dest address	proto	source port	dest port	flag bit	check conxion
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	any	
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK	X
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	---	
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	----	X
deny	all	all	all	all	all	all	

# Application gateways

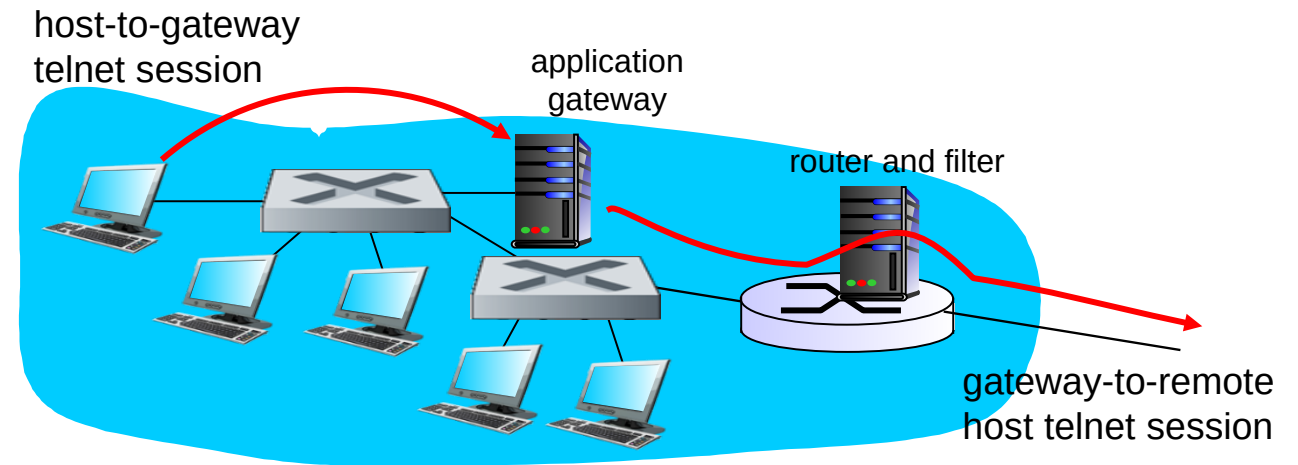
- filters packets on application data as well as on IP/TCP/UDP fields.
- example:* allow select internal users to telnet outside.



1. require all telnet users to telnet through gateway.
2. for authorized users, gateway sets up telnet connection to dest host. Gateway relays data between 2 connections
3. router filter blocks all telnet connections not originating from gateway.

# Application gateways

- filter packets on application data as well as on IP/TCP/UDP fields.
- *example*: allow select internal users to telnet outside



1. require all telnet users to telnet through gateway.
2. for authorized users, gateway sets up telnet connection to dest host. Gateway relays data between 2 connections
3. router filter blocks all telnet connections not originating from gateway.

# Limitations of firewalls, gateways

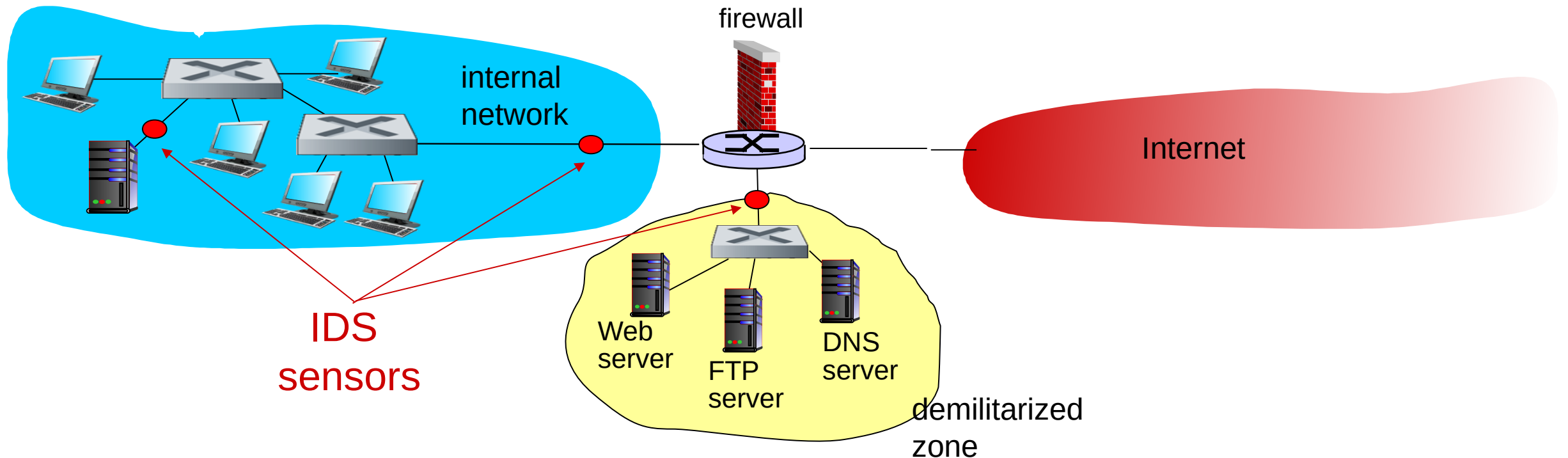
- *IP spoofing*: router can't know if data “really” comes from claimed source
- if multiple app's. need special treatment, each has own app. gateway
- client software must know how to contact gateway.
  - e.g., must set IP address of proxy in Web browser
- filters often use all or nothing policy for UDP
- *tradeoff*: degree of communication with outside world, level of security
- many highly protected sites still suffer from attacks

# Intrusion detection systems

- **packet filtering:**
  - operates on TCP/IP headers only
  - no correlation check among sessions
- *IDS: intrusion detection system*
  - *deep packet inspection*: look at packet contents (e.g., check character strings in packet against database of known virus, attack strings)
  - *examine correlation* among multiple packets
    - port scanning
    - network mapping
    - DoS attack

# Intrusion detection systems

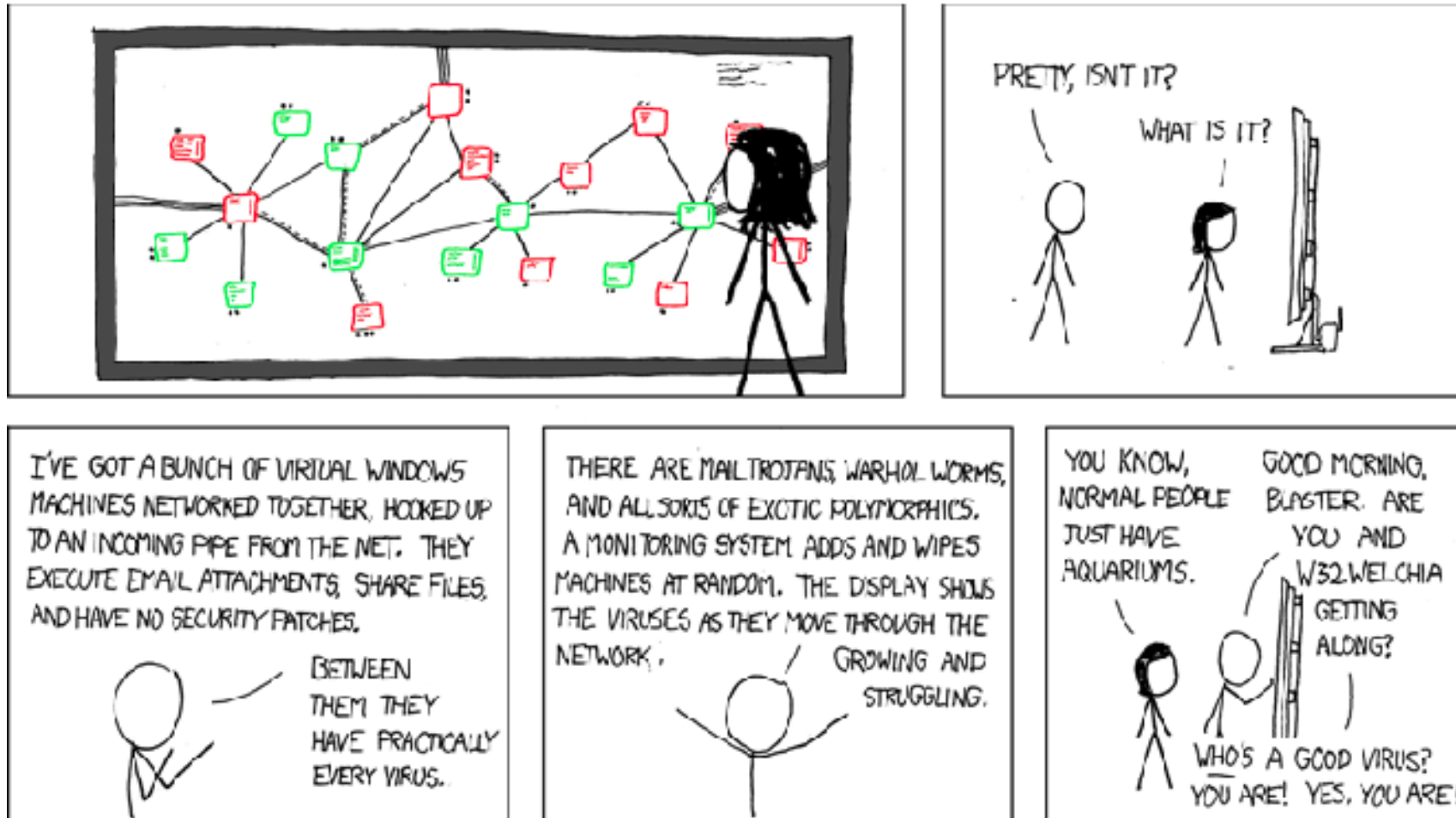
- multiple IDSs: different types of checking at different locations





# Honeypots

- mechanism set to detect, deflect unauthorized use of systems.



# Network Security (summary)

## basic techniques.....

- cryptography (symmetric and public)
- message integrity
- end-point authentication

## .... used in many different security scenarios

- secure email
- secure transport (SSL)
- IP sec
- 802.11

## operational security: firewalls and IDS

# Review



- Posters – Due Friday, Nov 29<sup>th</sup>, 5PM Central
  - No extensions and no exceptions – if it's not submitted to ilearn by 5PM Central, it will not be accepted
- Template: Search for (a0/a1 poster template), pick one you like
- Monday 11/18: Quiz, ungraded, open book, open laptop (1:25-2:20PM)
- Wednesday 11/20: Class evaluation (bring your tablets/laptops/phones) – Last 10 Minutes
  - <https://tntech.campuslabs.com/courseeval/>