

# Chapter 8: Network Security

## *Chapter goals:*

- understand principles of network security:
  - cryptography and its *many* uses beyond “confidentiality”
  - authentication
  - message integrity
- security in practice:
  - firewalls and intrusion detection systems
  - security in application, transport, network, link layers

# Chapter 8 roadmap

*8.1 What is network security?*

8.2 Principles of cryptography

8.3 Message integrity, authentication

~~8.4 Securing e-mail~~

8.5 Securing TCP connections: SSL

8.6 Network layer security: IPsec

~~8.7 Securing wireless LANs~~

8.8 Operational security: firewalls and IDS

# What is network security?

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

- sender encrypts message
- receiver decrypts message

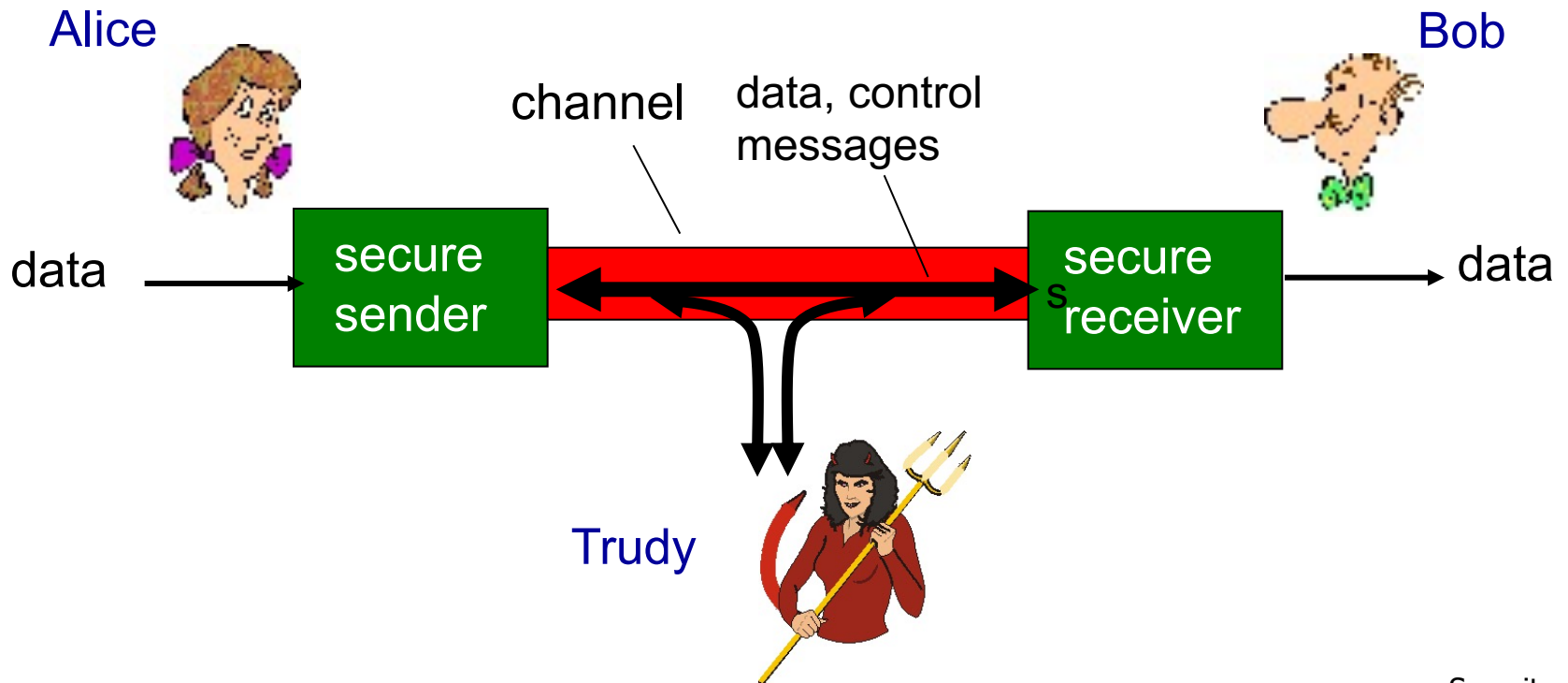
***authentication:*** sender, receiver want to confirm identity of each other

***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

- well-known in network security world
- Bob, Alice (lovers!) want to communicate “securely”
- Trudy (intruder) may intercept, delete, add messages



# There are bad guys (and girls) out there!

Q: What can a “bad guy” do?

A: A lot! See section 1.6

- *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)

# Chapter 8 roadmap

8.1 What is network security?

8.2 *Principles of cryptography*

8.3 Message integrity, authentication

~~8.4 Securing e-mail~~

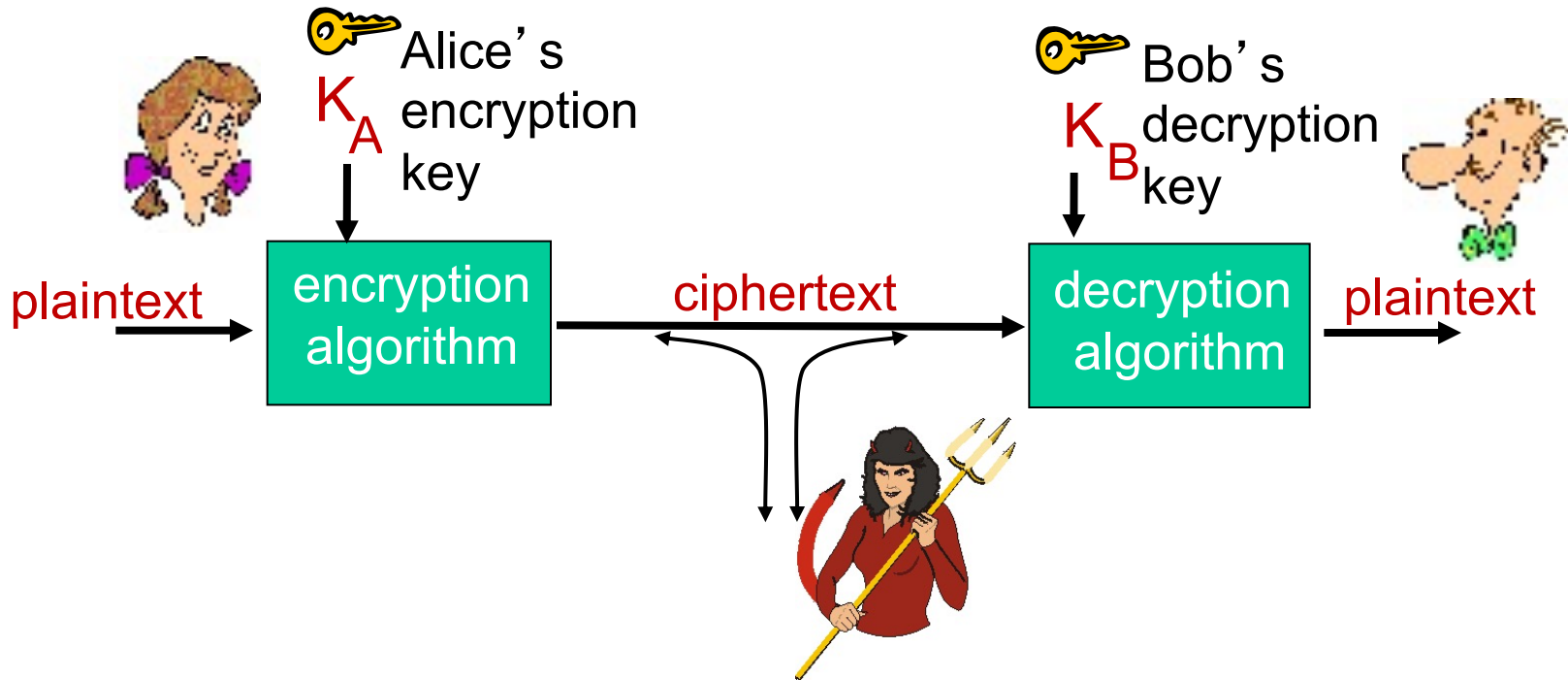
8.5 Securing TCP connections: SSL

8.6 Network layer security: IPsec

~~8.7 Securing wireless LANs~~

8.8 Operational security: firewalls and IDS

# The language of cryptography

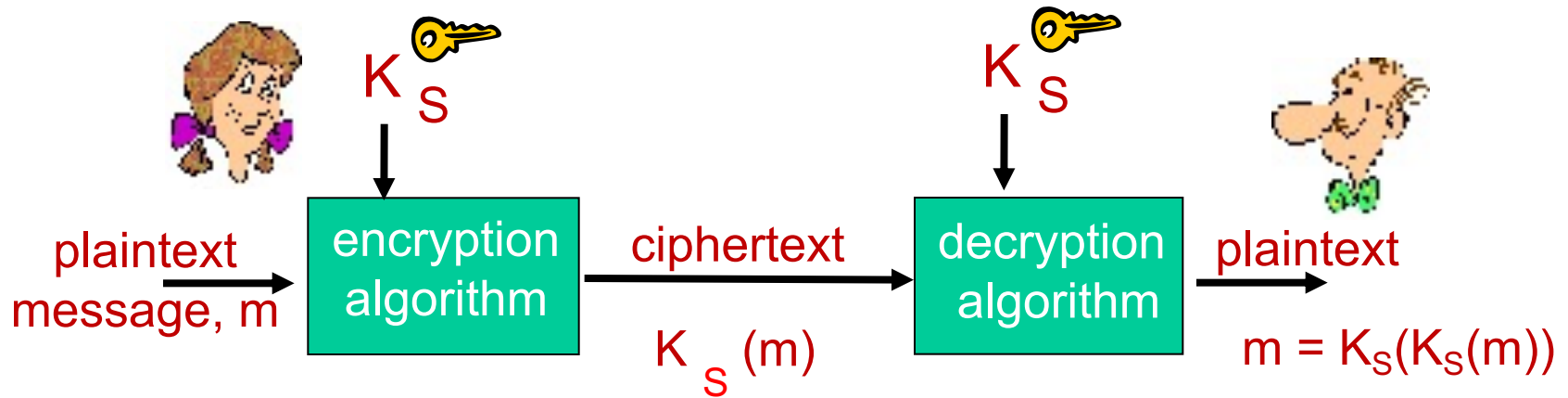


$m$  plaintext message

$K_A(m)$  ciphertext, encrypted with key  $K_A$

$m = K_B(K_A(m))$

# Symmetric key cryptography



**symmetric key crypto:** Bob and Alice share same (symmetric) key:  $K_S$

- e.g., key is knowing substitution pattern in mono alphabetic substitution cipher

**Q:** how do Bob and Alice agree on key value?



# AES: Advanced Encryption Standard

- symmetric-key NIST standard (Nov 2001)
- processes data in 128 bit blocks
- 128, 192, or 256 bit keys
- brute force decryption (try each key) taking 1 sec on DES, takes 149 trillion years for AES

# Public Key Cryptography



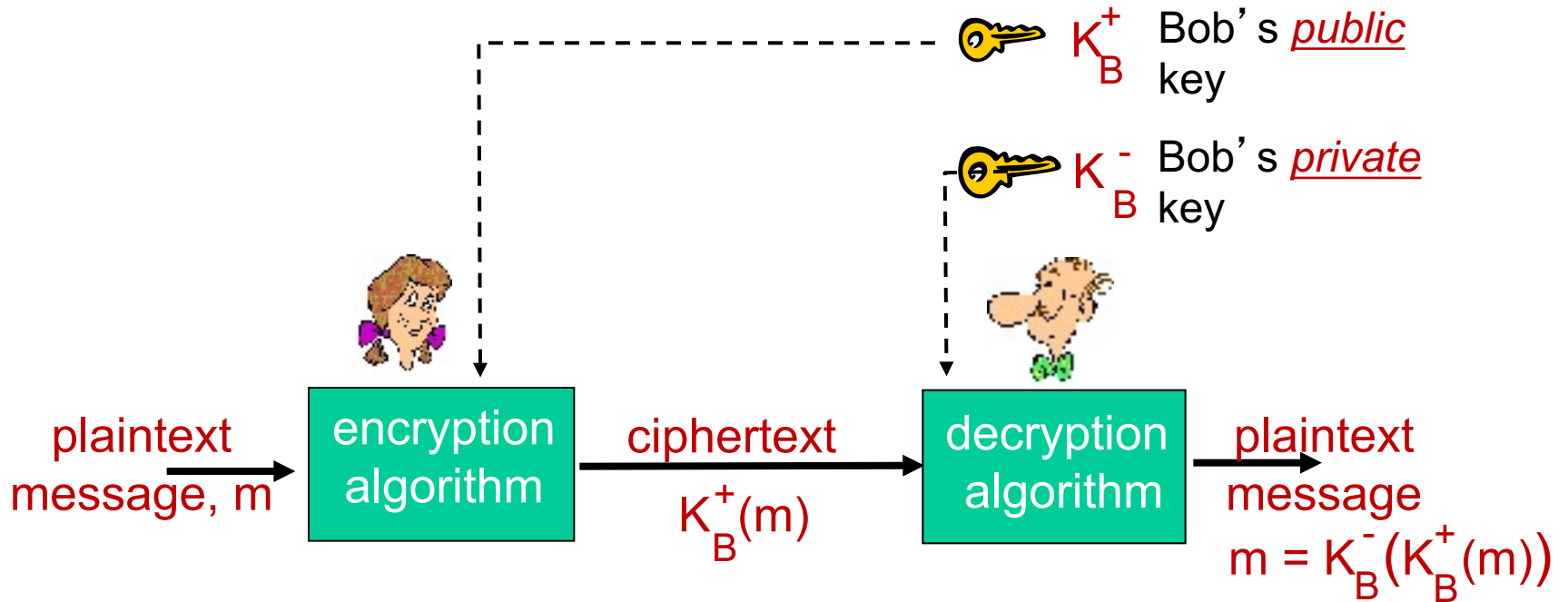
## *symmetric key crypto*

- requires sender, receiver know shared secret key
- Q: how to agree on key in first place (particularly if never “met”)?

## *public key crypto*

- radically different approach [Diffie-Hellman76, RSA78]
- sender, receiver do *not* share secret key
- *public* encryption key known to *all*
- *private* decryption key known only to receiver

# Public key cryptography



# Chapter 8 roadmap

8.1 What is network security?

8.2 Principles of cryptography

8.3 Message integrity, *authentication*

~~8.4 Securing e-mail~~

8.5 Securing TCP connections: SSL

8.6 Network layer security: IPsec

~~8.7 Securing wireless LANs~~

8.8 Operational security: firewalls and IDS

# Authentication

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

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



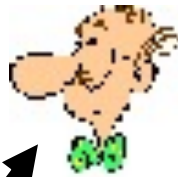
Failure scenario??



# Authentication

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

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

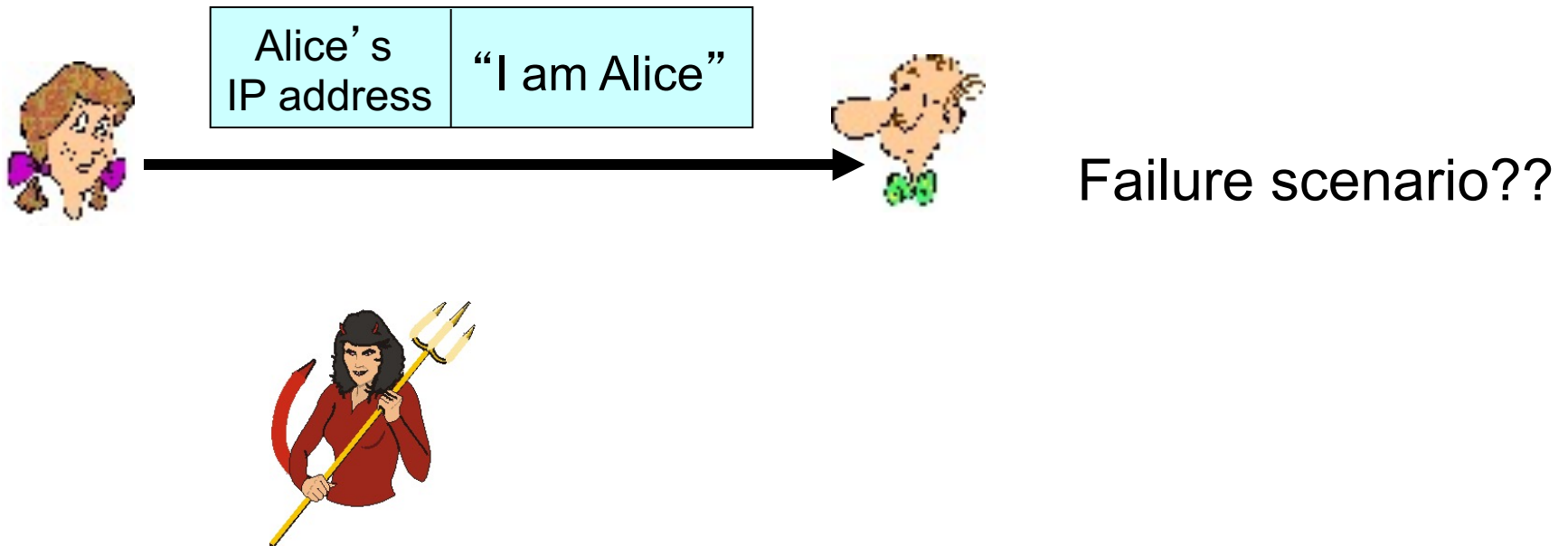


“I am Alice”

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

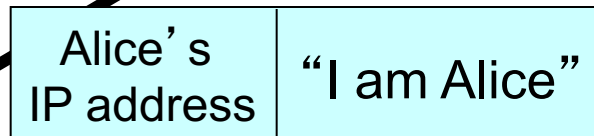
# Authentication: another try

*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

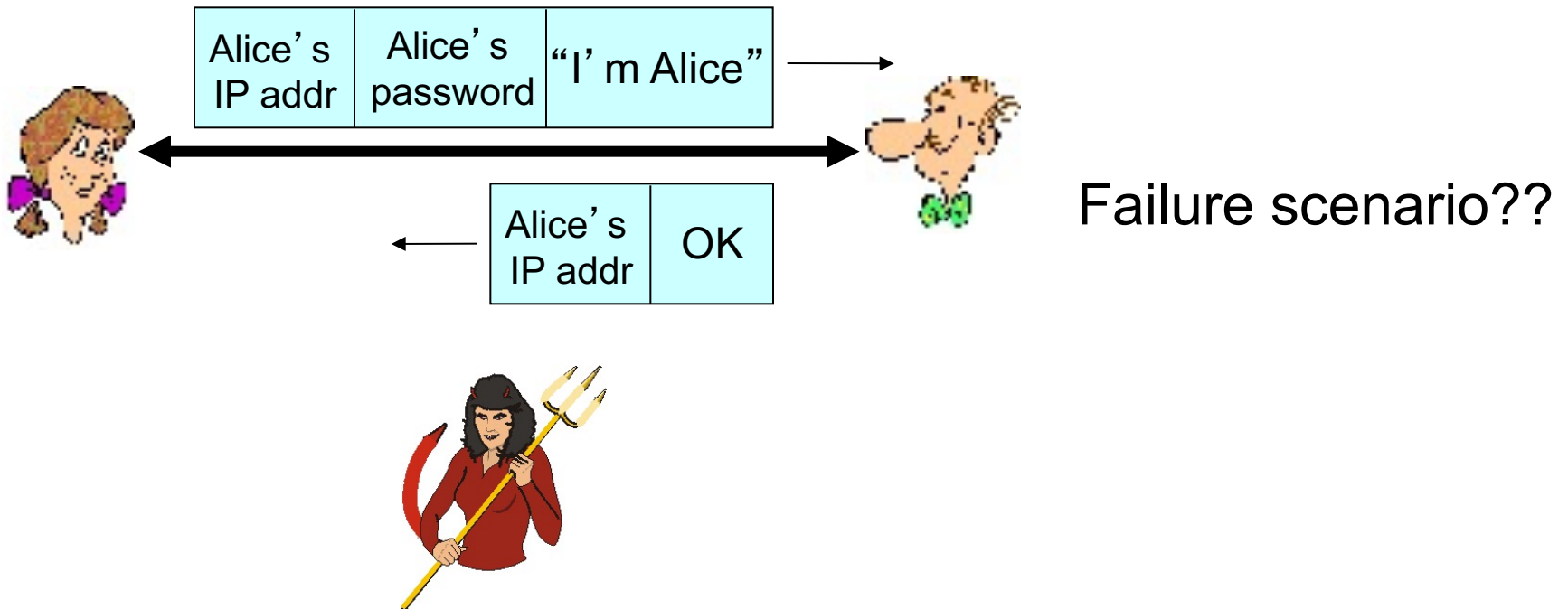


Trudy can create a packet  
“spoofing”  
Alice's address



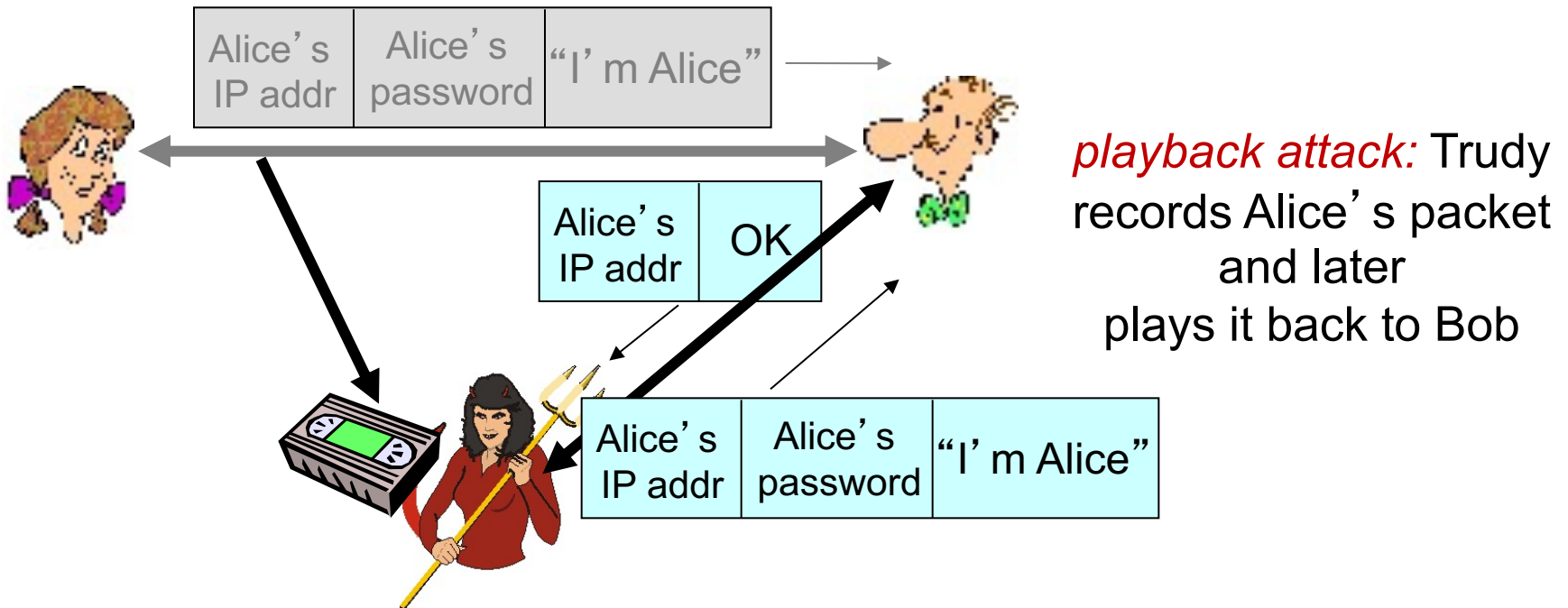
# Authentication: another try

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



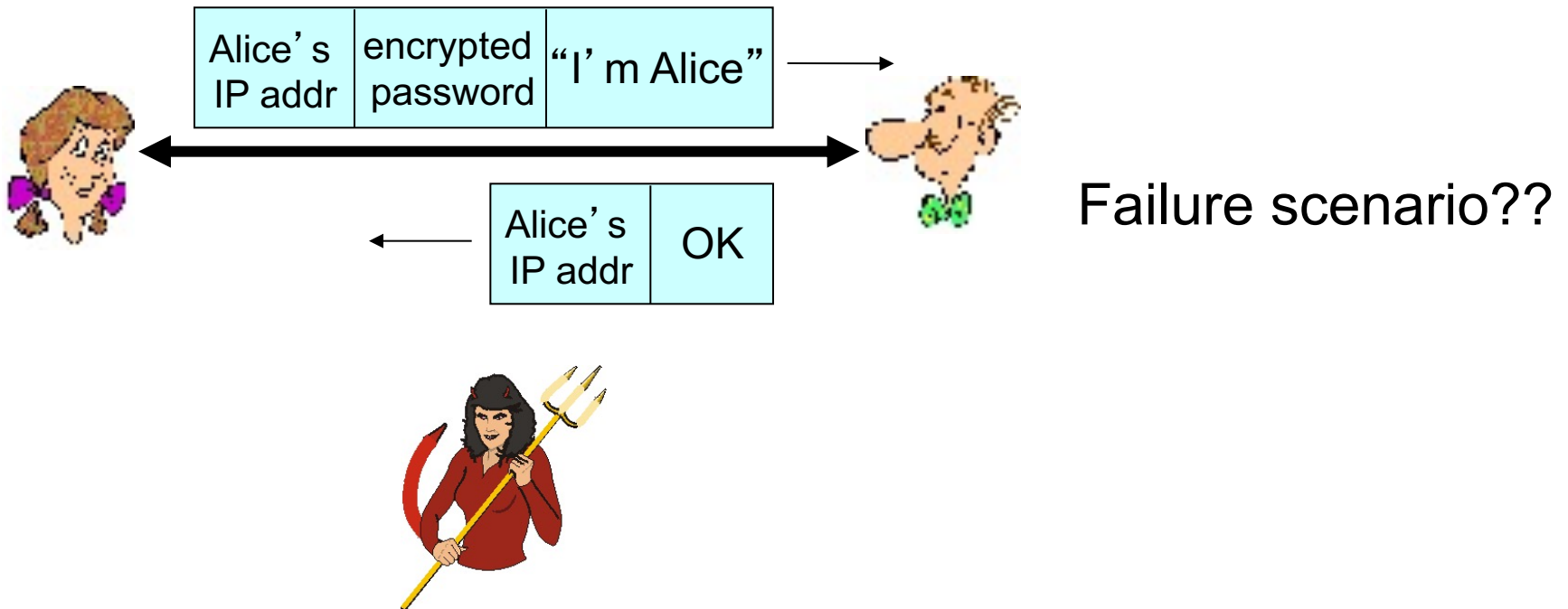
# Authentication: another try

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



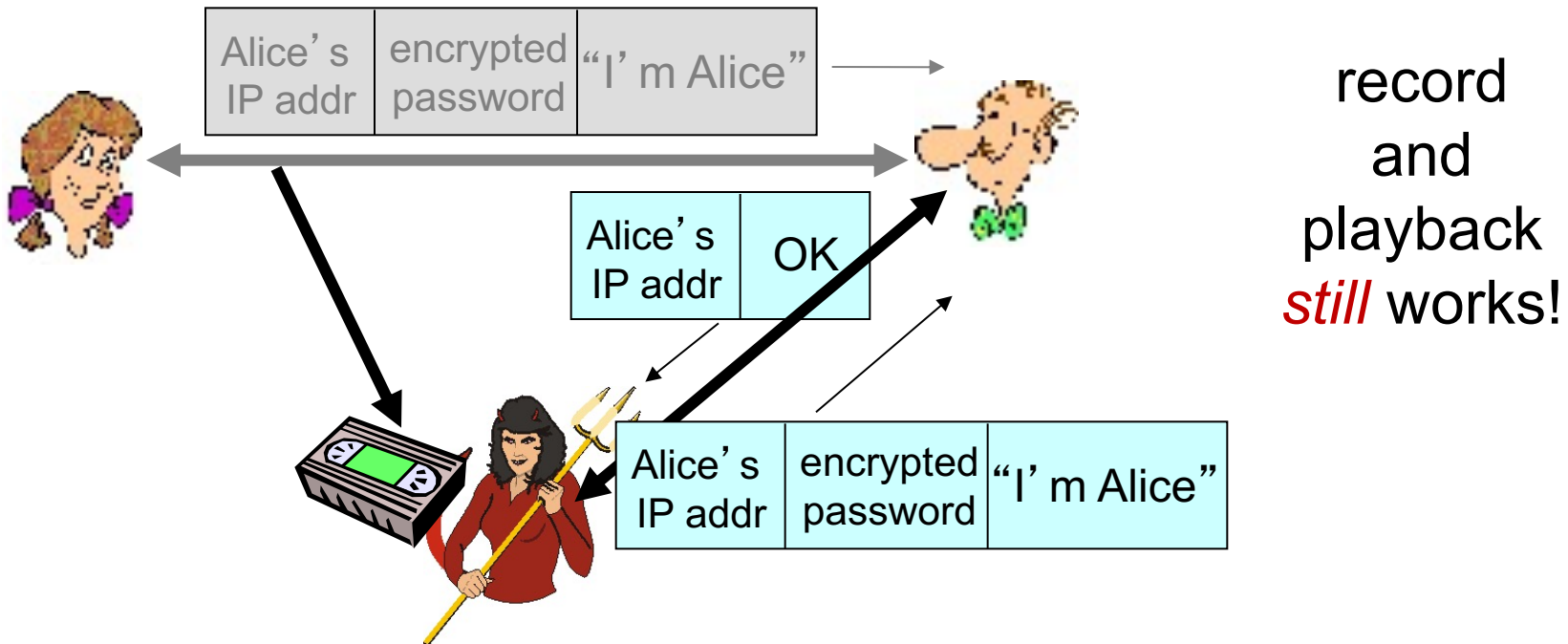
# Authentication: yet another try

*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.

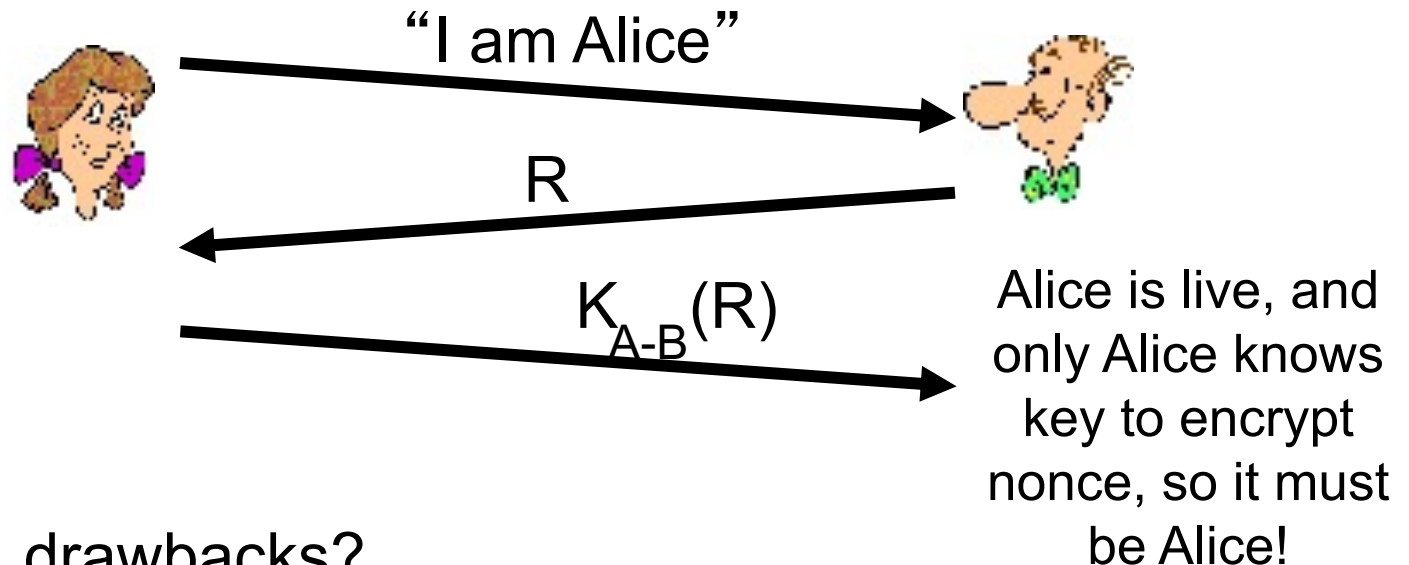


# Authentication: yet another try

**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



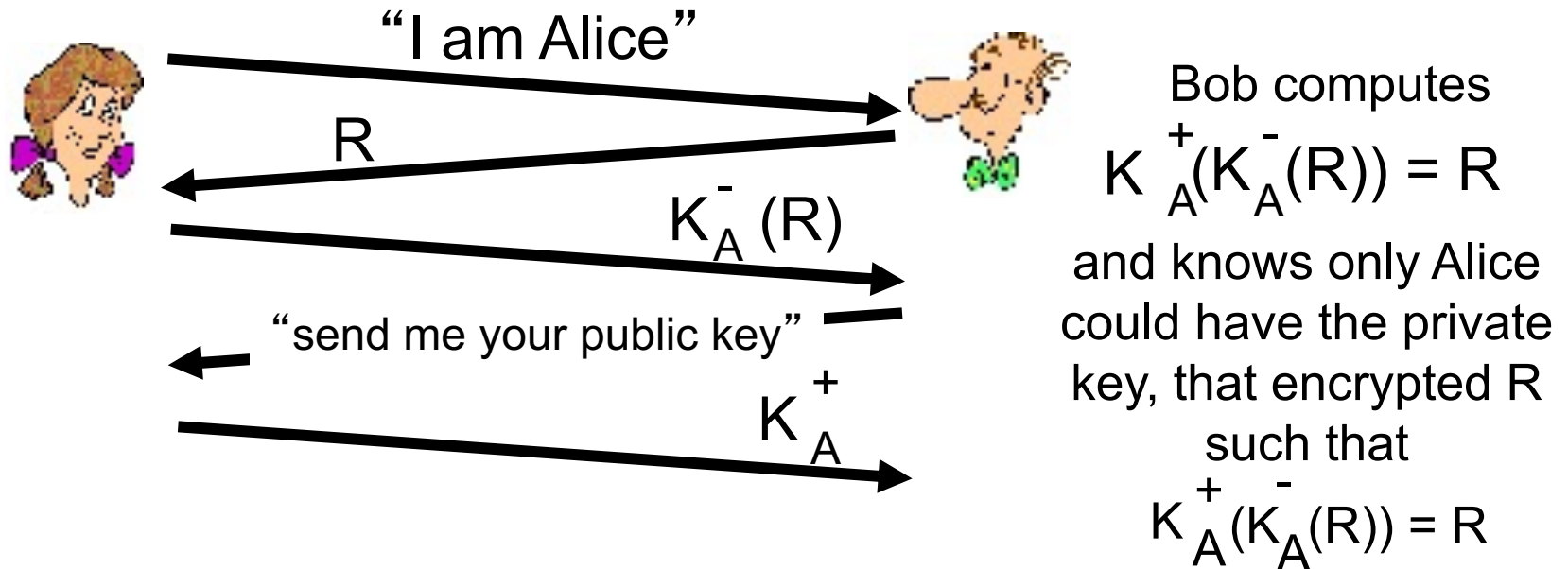
Failures, drawbacks?

# Authentication: ap5.0

ap4.0 requires shared symmetric key

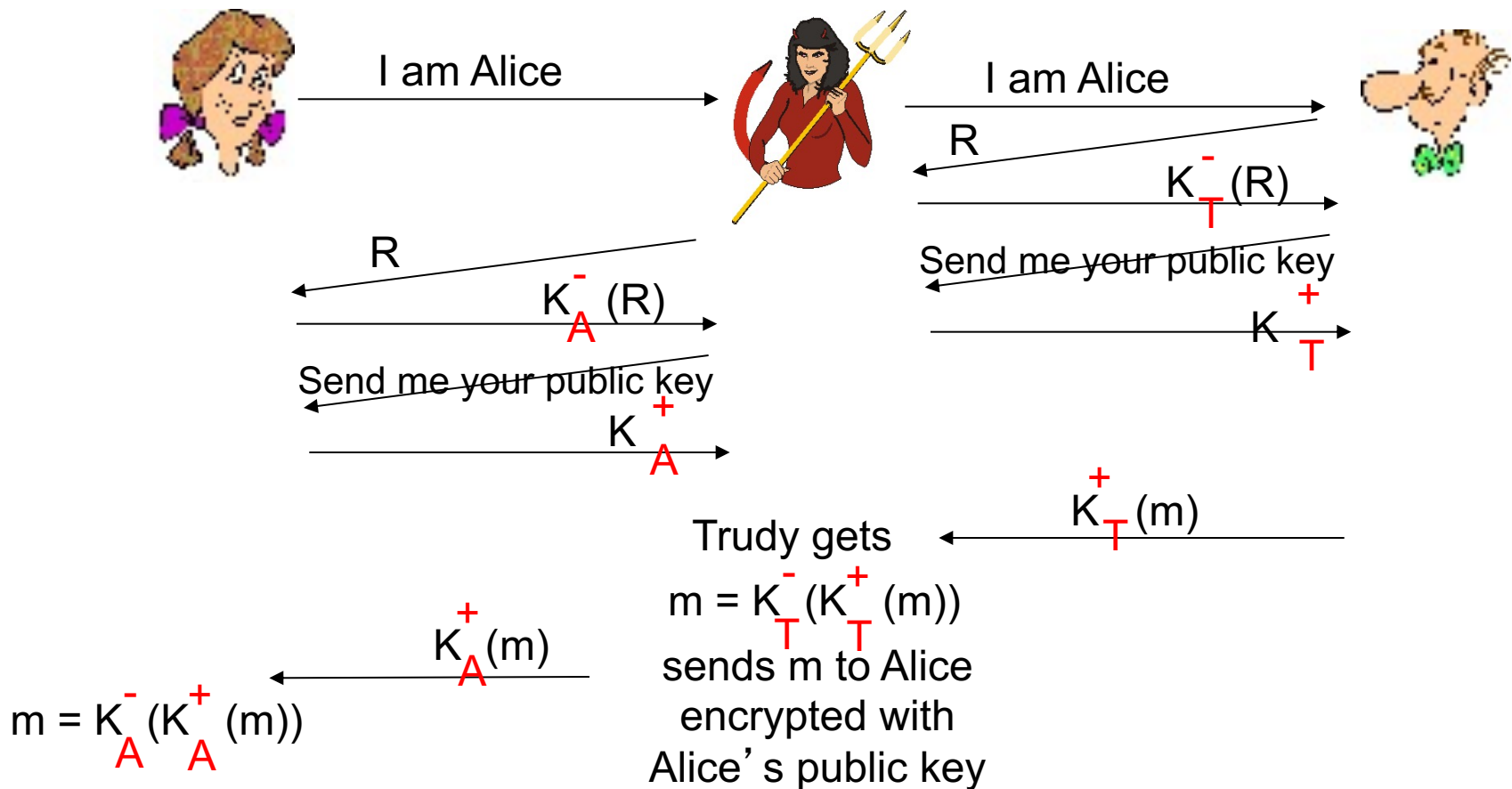
- can we authenticate using public key techniques?

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



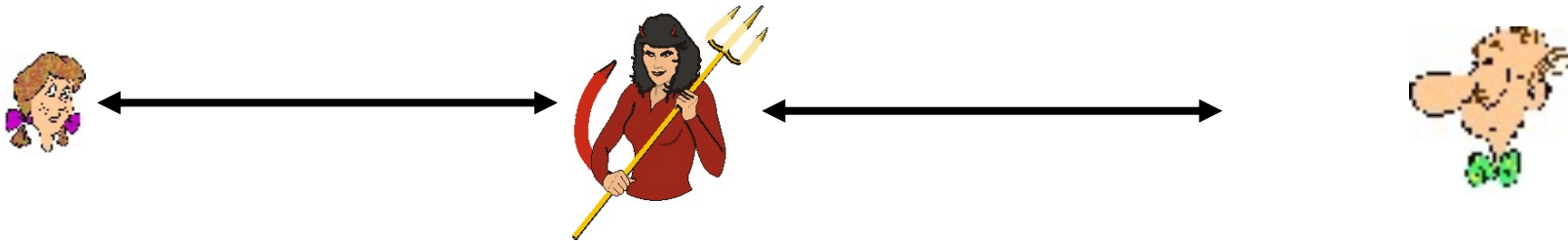
# ap5.0: security hole

*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. (e.g., so Bob, Alice can meet one week later and recall conversation!)
- problem is that Trudy receives all messages as well!



# Chapter 8 roadmap

8.1 What is network security?

8.2 Principles of cryptography

8.3 *Message integrity*, authentication

~~8.4 Securing e-mail~~

8.5 Securing TCP connections: SSL

8.6 Network layer security: IPsec

~~8.7 Securing wireless LANs~~

8.8 Operational security: firewalls and IDS

# Digital signatures

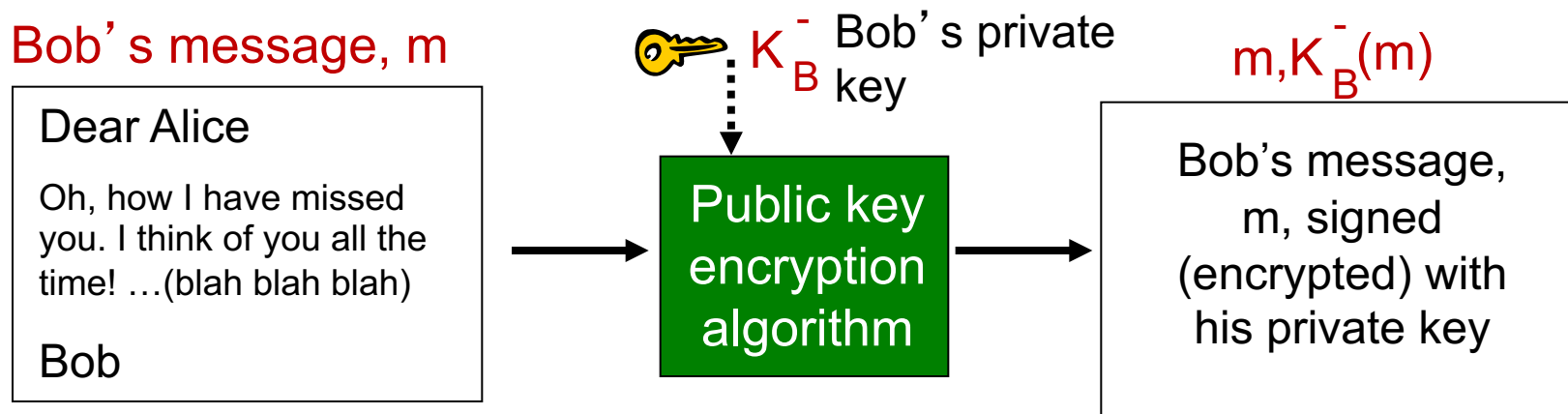
cryptographic technique analogous to hand-written signatures:

- sender (Bob) digitally signs document, establishing he is document owner/creator.
- *verifiable, nonforgeable*: recipient (Alice) can prove to someone that Bob, and no one else (including Alice), must have signed document

# Digital signatures

simple digital signature for message  $m$ :

- Bob signs  $m$  by encrypting with his private key  $K_B^-$ , creating “signed” message,  $K_B^-(m)$

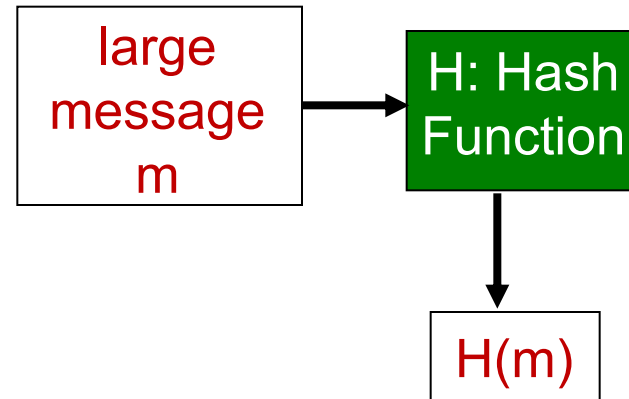


# Message digests

computationally  
expensive to public-key-  
encrypt long messages

**goal:** fixed-length, easy-  
to-compute digital  
“fingerprint”

- apply hash function  $H$  to  $m$ , get fixed size message digest,  $H(m)$ .

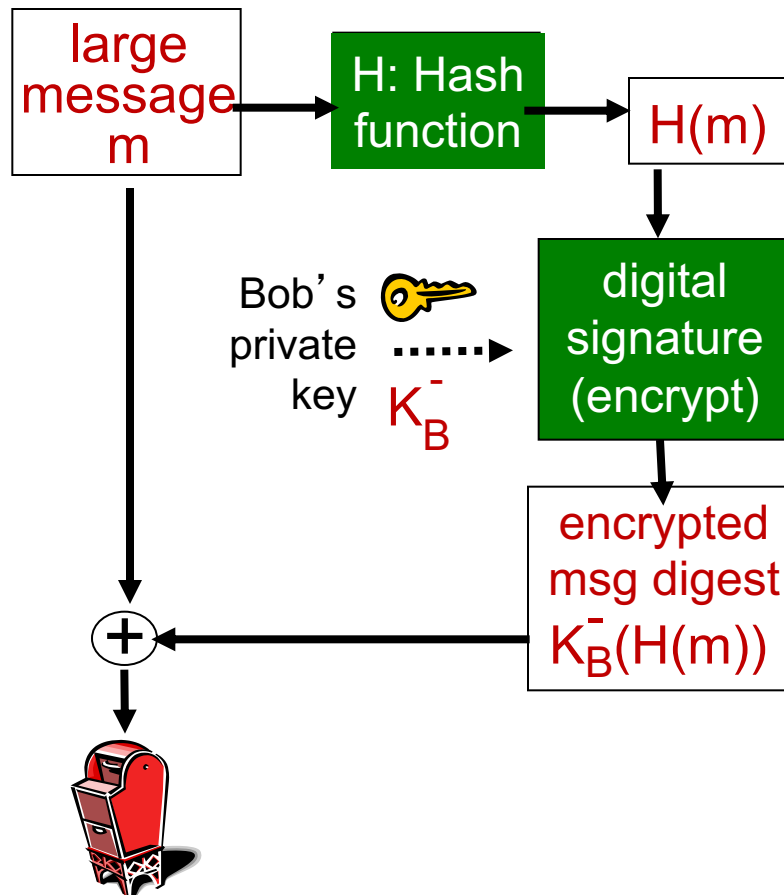


**Hash function properties:**

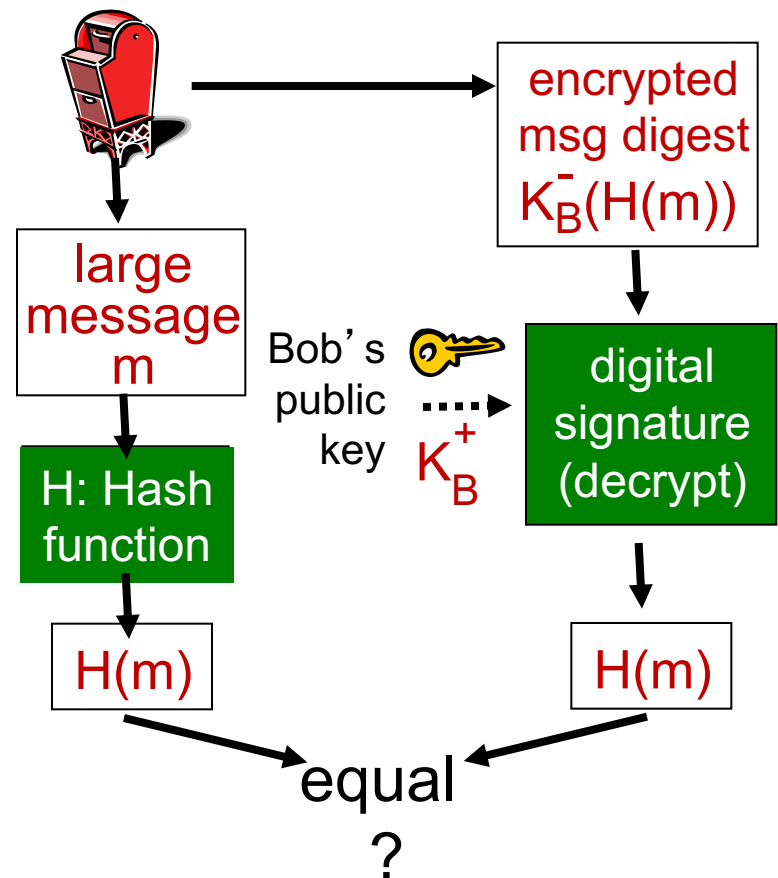
- many-to-1
- produces fixed-size msg digest (fingerprint)
- given message digest  $x$ , computationally infeasible to find  $m$  such that  $x = H(m)$

# Digital signature = signed message digest

Bob sends digitally signed message:



Alice verifies signature, integrity of digitally signed message:

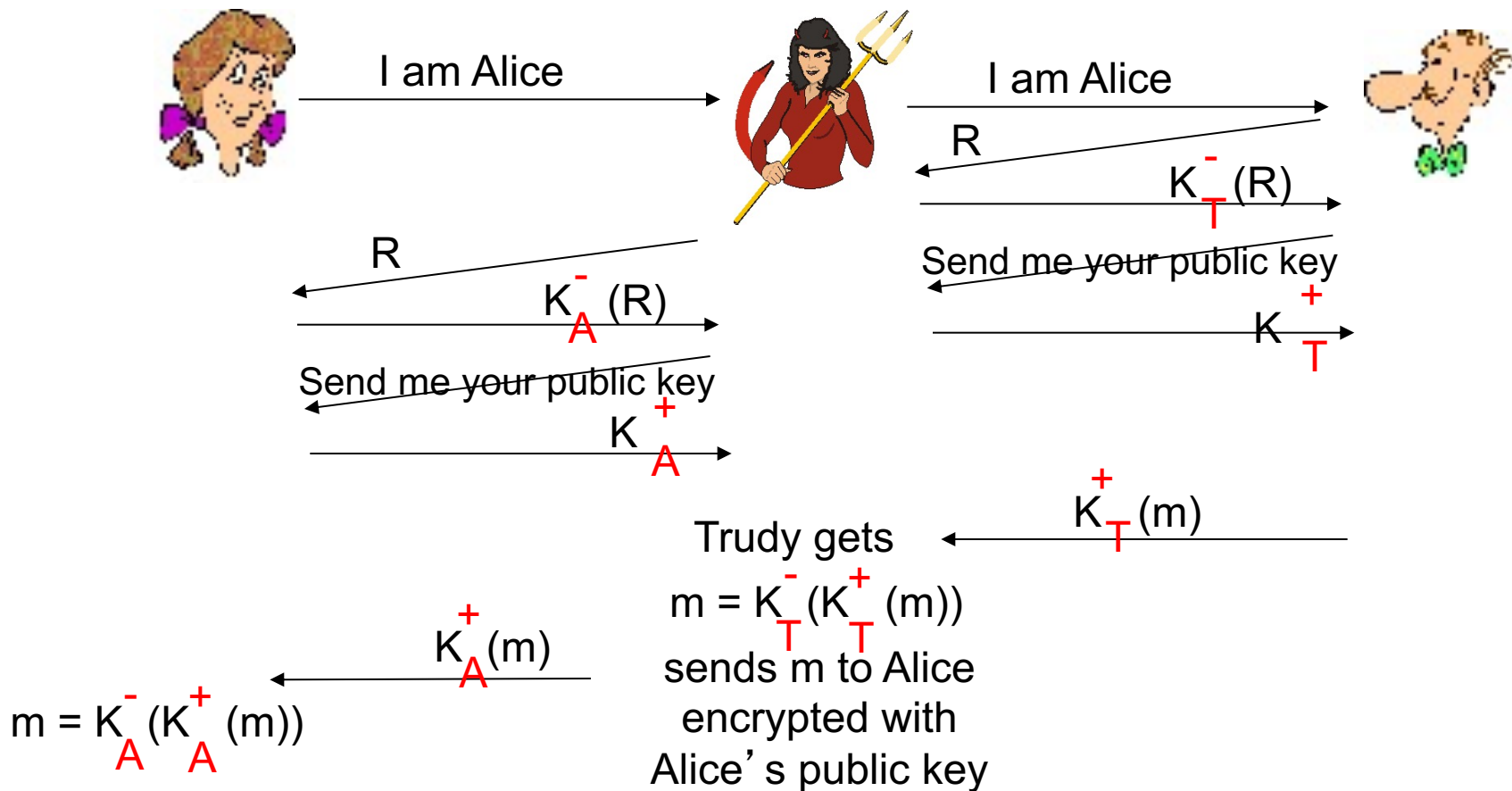


# Hash function algorithms

- **MD5 hash function widely used (RFC 1321)**
  - computes 128-bit message digest in 4-step process.
  - arbitrary 128-bit string  $x$ , appears difficult to construct msg  $m$  whose MD5 hash is equal to  $x$
- **SHA-1 is also used**
  - US standard [NIST, FIPS PUB 180-1]
  - 160-bit message digest

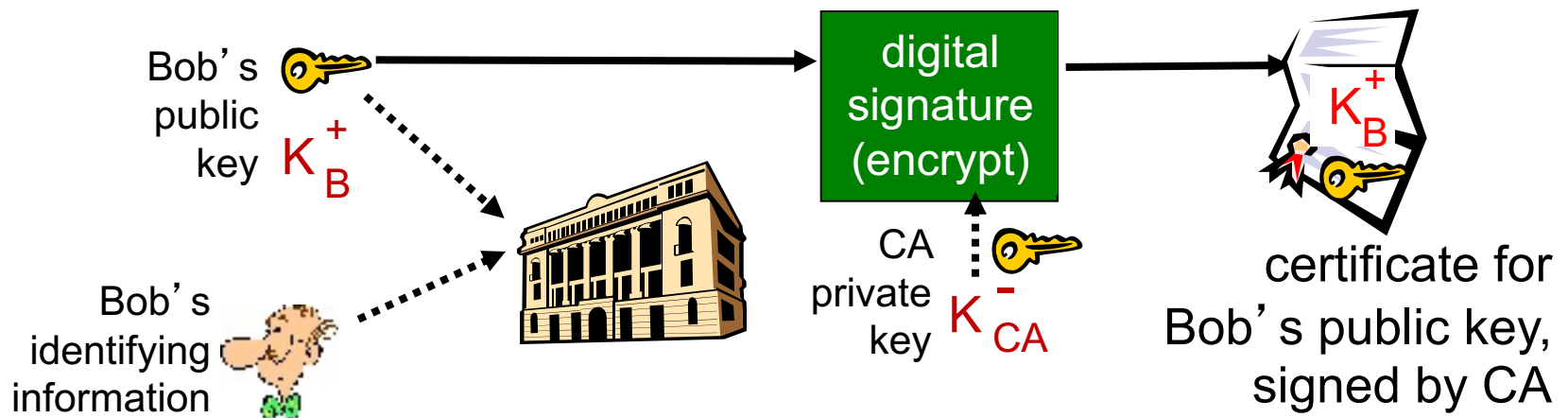
# Recall: ap5.0 security hole

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



# Certification authorities

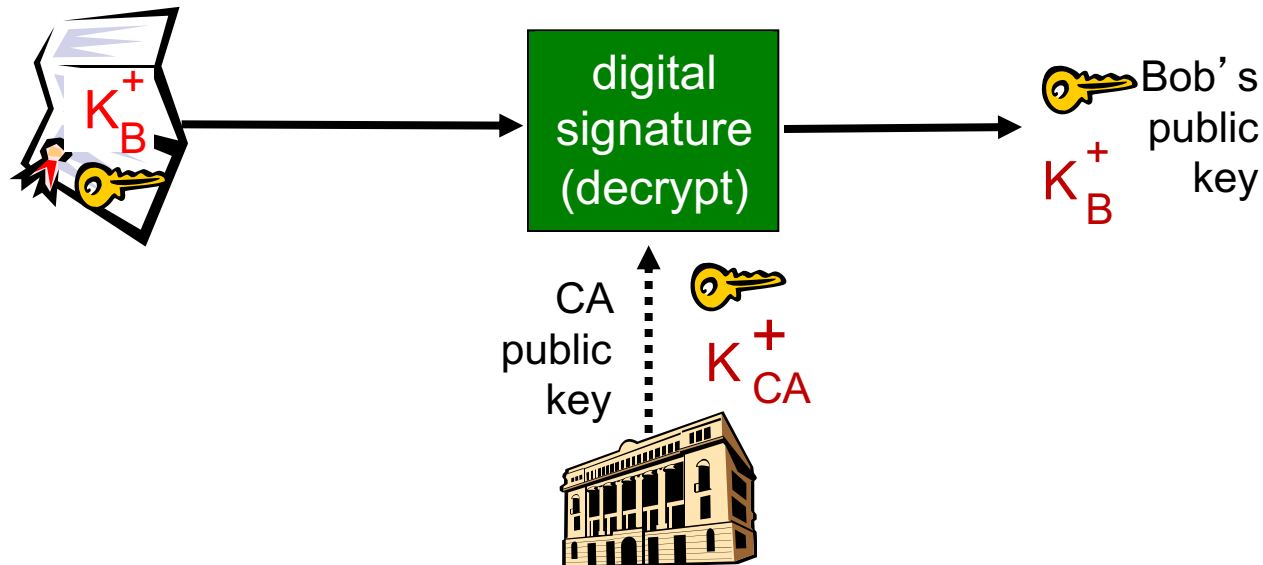
- **certification authority (CA):** binds public key to particular entity, E.
- E (person, router) registers its public key with CA.
  - E provides “proof of identity” to CA.
  - CA creates certificate binding E to its public key.
  - certificate containing E’s public key digitally signed by CA – CA says “this is E’s public key”





# Certification authorities

- when Alice wants Bob's public key:
  - gets Bob's certificate (Bob or elsewhere).
  - apply CA's public key to Bob's certificate, get Bob's public key



# Chapter 8 roadmap

8.1 What is network security?

8.2 Principles of cryptography

8.3 Message integrity

~~8.4 Securing e-mail~~

*8.5 Securing TCP connections: SSL*

8.6 Network layer security: IPsec

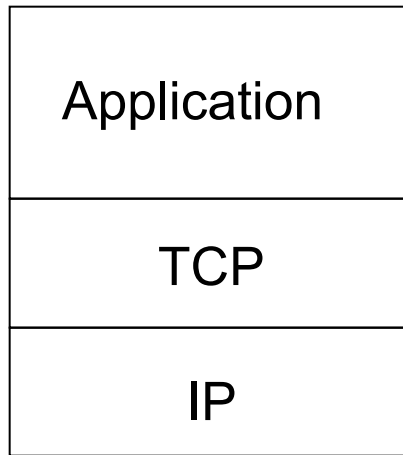
~~8.7 Securing wireless LANs~~

8.8 Operational security: firewalls and IDS

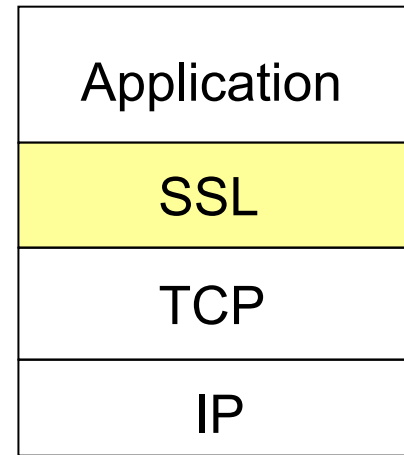
# SSL: Secure Sockets Layer

- 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*

# 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

# SSL cipher suite

- cipher suite
  - public-key algorithm
  - symmetric encryption algorithm
  - MAC algorithm
- SSL supports several cipher suites
- negotiation: client, server agree on cipher suite
  - client offers choice
  - server picks one

## common SSL symmetric ciphers

- DES – Data Encryption  
Standard: block
- 3DES – Triple strength: block
- RC2 – Rivest Cipher 2: block
- RC4 – Rivest Cipher 4: stream

## SSL Public key encryption

- RSA

# Real SSL: handshake (I)

## *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: handshaking (3)

last 2 steps protect handshake from tampering

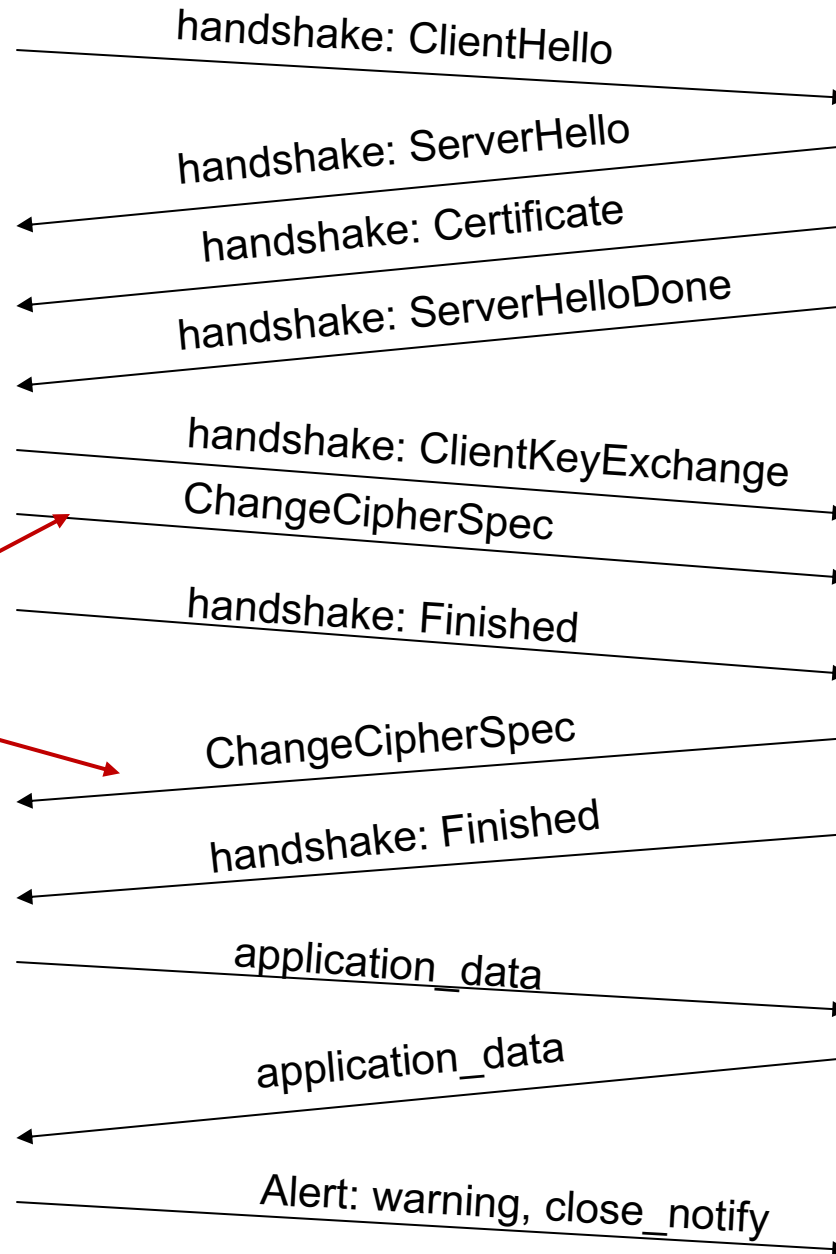
- client typically offers range of algorithms, some strong, some weak
- man-in-the middle could delete stronger algorithms from list
- last 2 steps prevent this
  - last two messages are encrypted



# Real SSL connection

*everything  
henceforth  
is encrypted*

TCP FIN follows



# Chapter 8 roadmap

8.1 What is network security?

8.2 Principles of cryptography

8.3 Message integrity

~~8.4 Securing e-mail~~

8.5 Securing TCP connections: SSL

*8.6 Network layer security: IPsec*

~~8.7 Securing wireless LANs~~

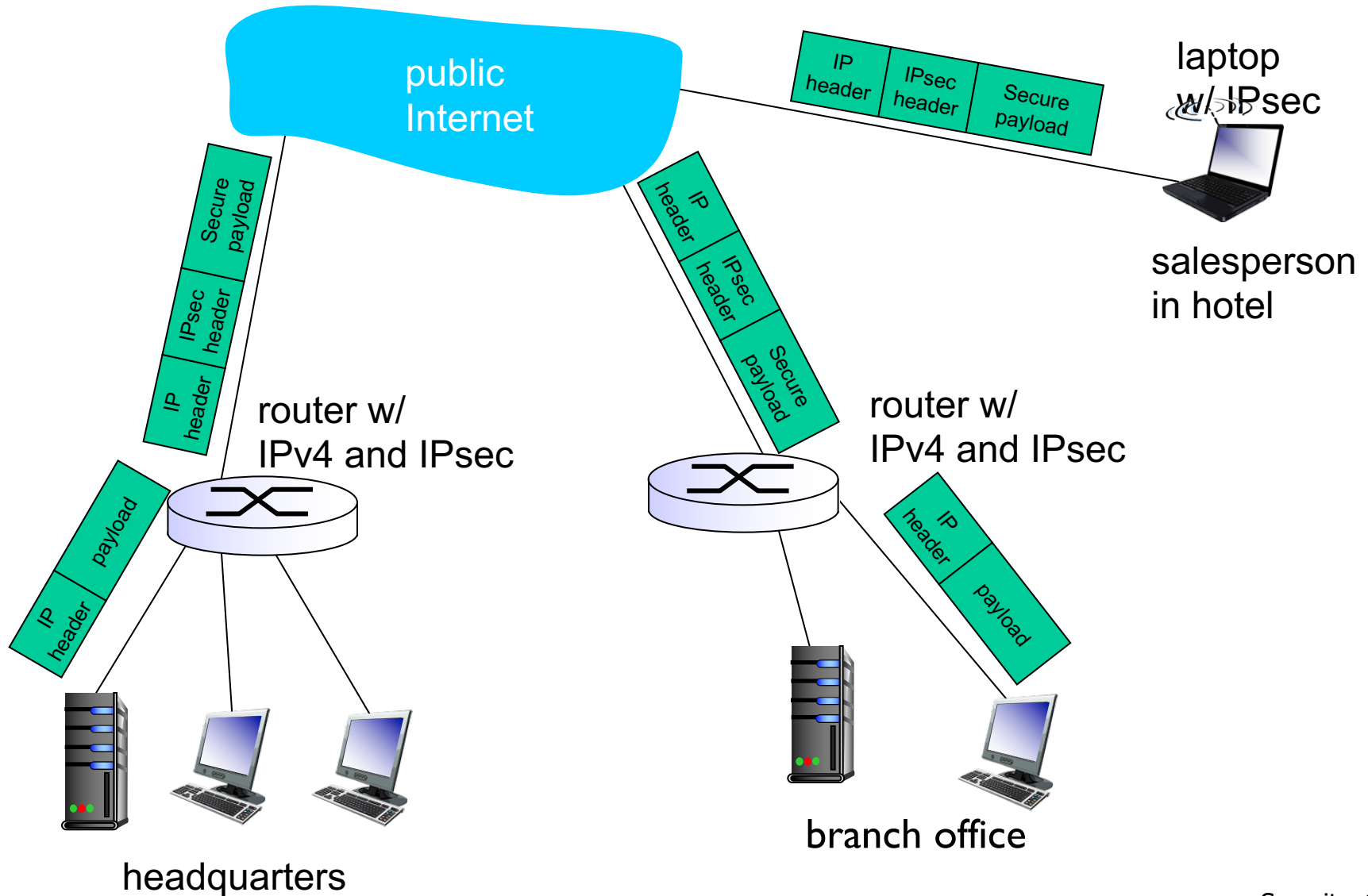
8.8 Operational security: firewalls and IDS

# 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

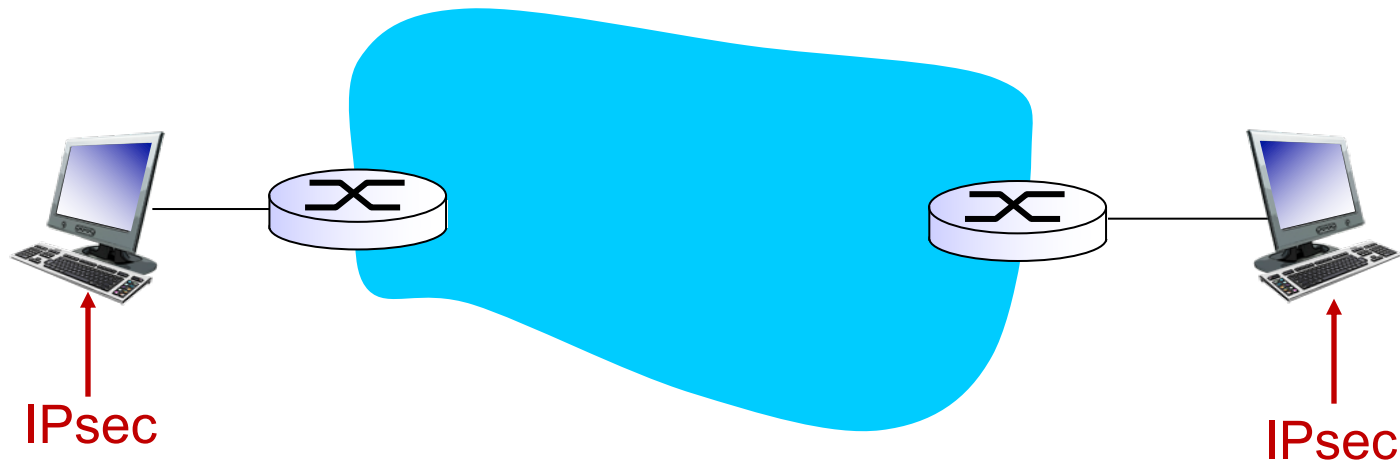
# Virtual Private Networks (VPNs)



# IPsec services

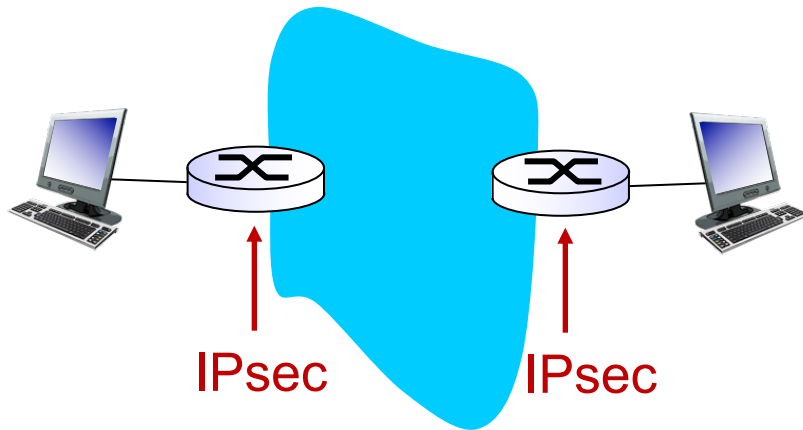
- data integrity
  - origin authentication
  - replay attack prevention
  - confidentiality
- 
- two protocols providing different service models:
    - AH
    - ESP

# IPsec transport mode

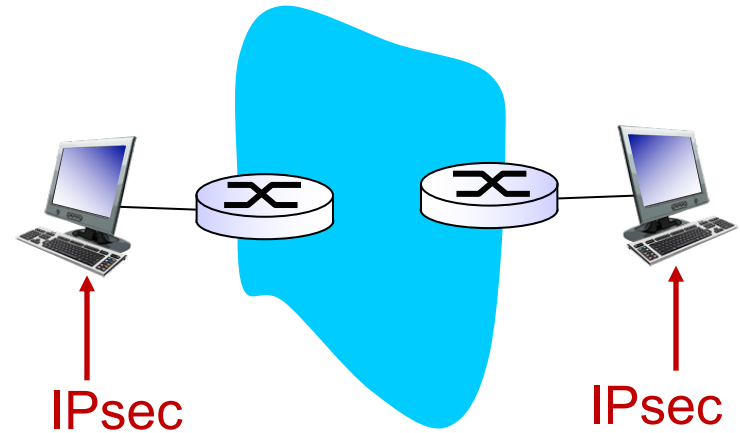


- IPsec datagram emitted and received by end-system
- protects upper level protocols

# 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

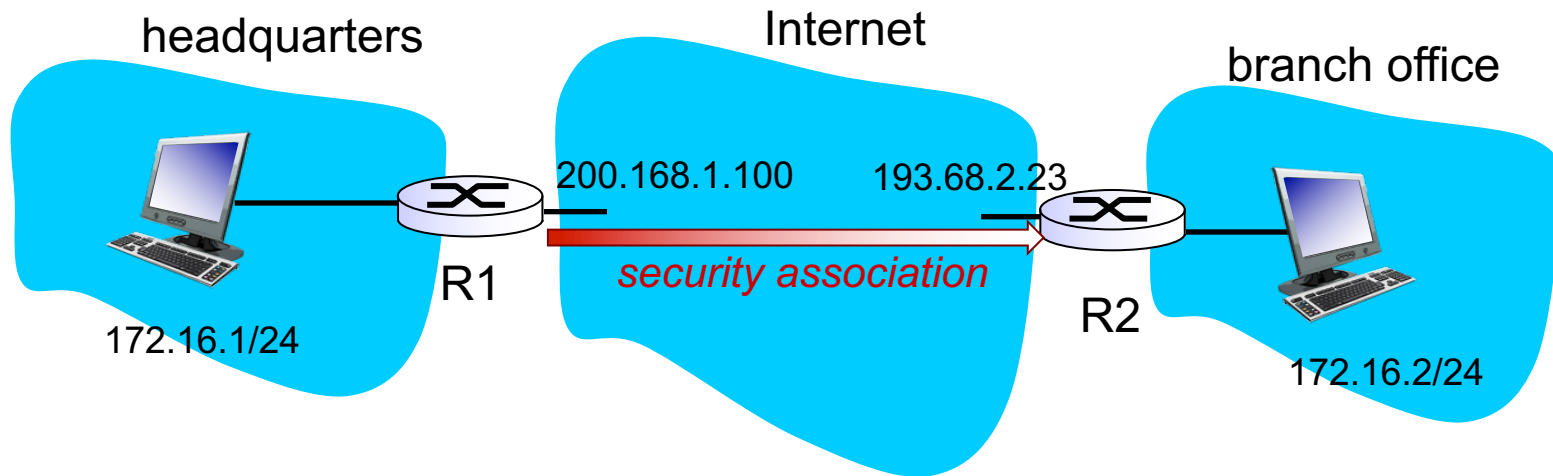


most common and  
most important

# Security associations (SAs)

- before sending data, “**security association (SA)**” established from sending to receiving entity
  - SAs are simplex: for only one direction
- ending, receiving entitles maintain *state information* about SA
  - recall: TCP endpoints also maintain state info
  - IP is connectionless; IPsec is connection-oriented!
- how many SAs in VPN w/ headquarters, branch office, and n traveling salespeople?

# Example SA from R1 to R2



## *R1 stores for SA:*

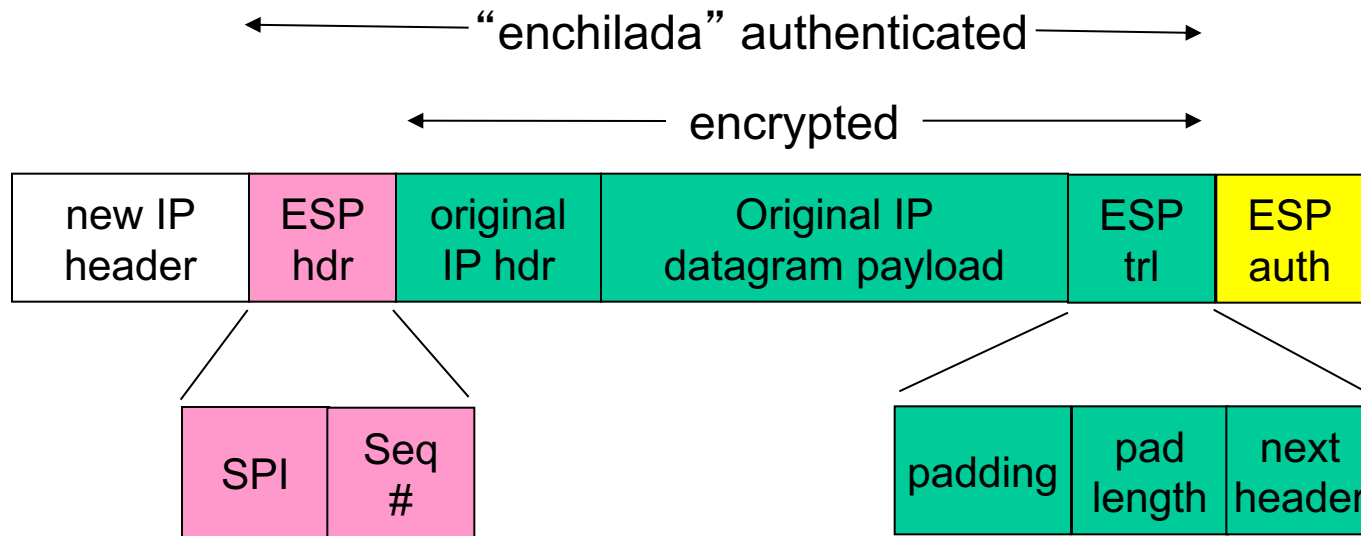
- 32-bit SA identifier: *Security Parameter Index (SPI)*
- origin SA interface (200.168.1.100)
- destination SA interface (193.68.2.23)
- type of encryption used (e.g., 3DES with CBC)
- encryption key
- type of integrity check used (e.g., HMAC with MD5)
- authentication key

# Security Association Database (SAD)

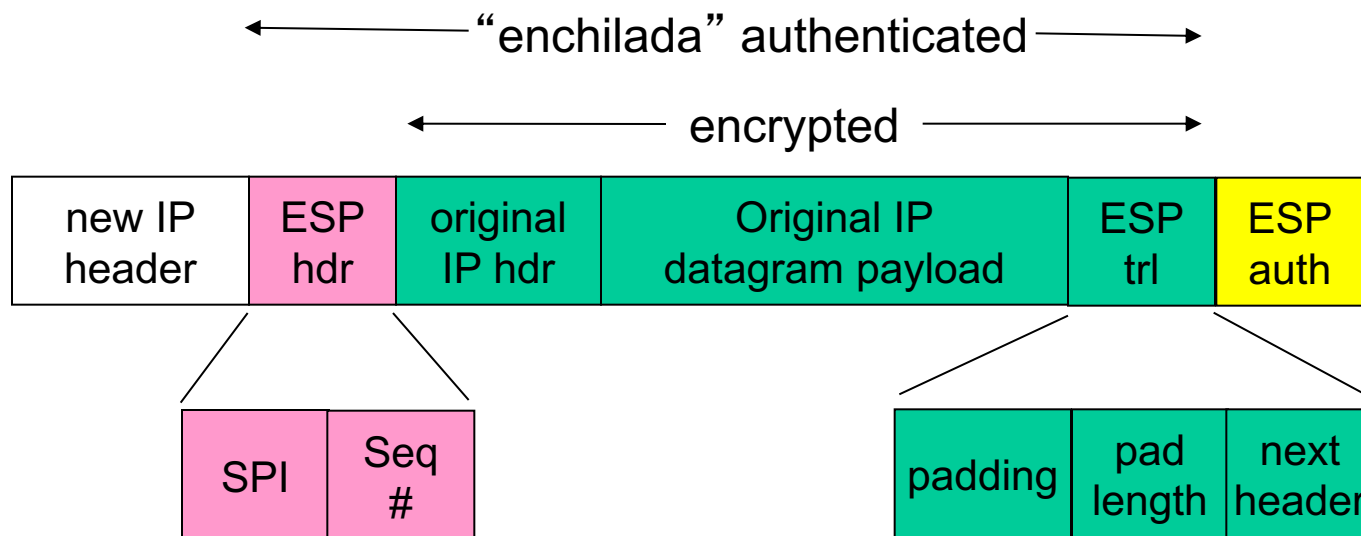
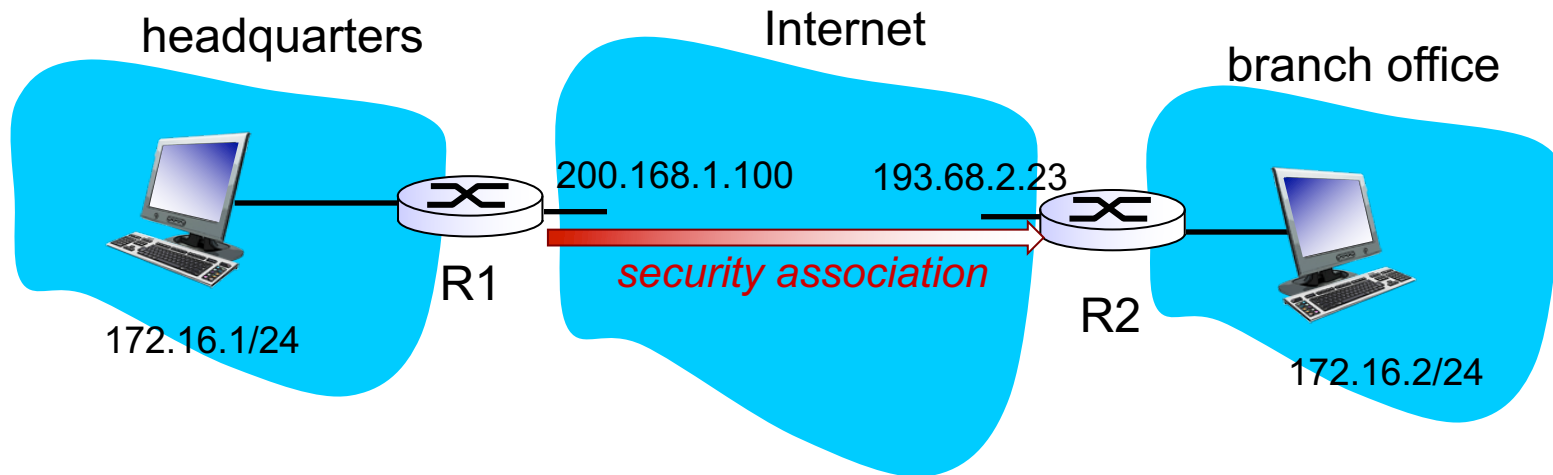
- endpoint holds SA state in *security association database (SAD)*, where it can locate them during processing.
- with  $n$  salespersons,  $2 + 2n$  SAs in R1's SAD
- when sending IPsec datagram, R1 accesses SAD to determine how to process datagram.
- when IPsec datagram arrives to R2, R2 examines SPI in IPsec datagram, indexes SAD with SPI, and processes datagram accordingly.

# IPsec datagram

focus for now on tunnel mode with ESP



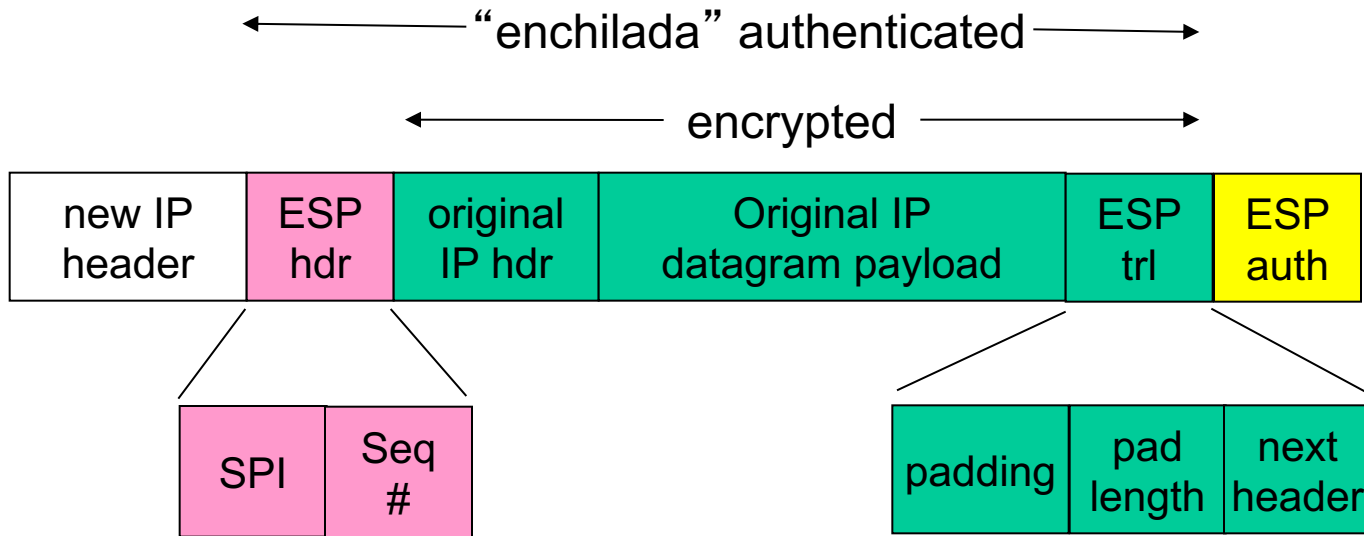
# What happens?



# RI: convert original datagram to IPsec datagram

- appends to back of original datagram (which includes original header fields!) an “ESP trailer” field.
- encrypts result using algorithm & key specified by SA.
- appends to front of this encrypted quantity the “ESP header, creating “enchilada”.
- creates authentication MAC over the *whole enchilada*, using algorithm and key specified in SA;
- appends MAC to back of enchilada, forming *payload*;
- creates brand new IP header, with all the classic IPv4 header fields, which it appends before payload

# Inside the enchilada:



- ESP trailer: Padding for block ciphers
- ESP header:
  - SPI, so receiving entity knows what to do
  - Sequence number, to thwart replay attacks
- MAC in ESP auth field is created with shared secret key



# Chapter 8 roadmap

8.1 What is network security?

8.2 Principles of cryptography

8.3 Message integrity

~~8.4 Securing e-mail~~

8.5 Securing TCP connections: SSL

8.6 Network layer security: IPsec

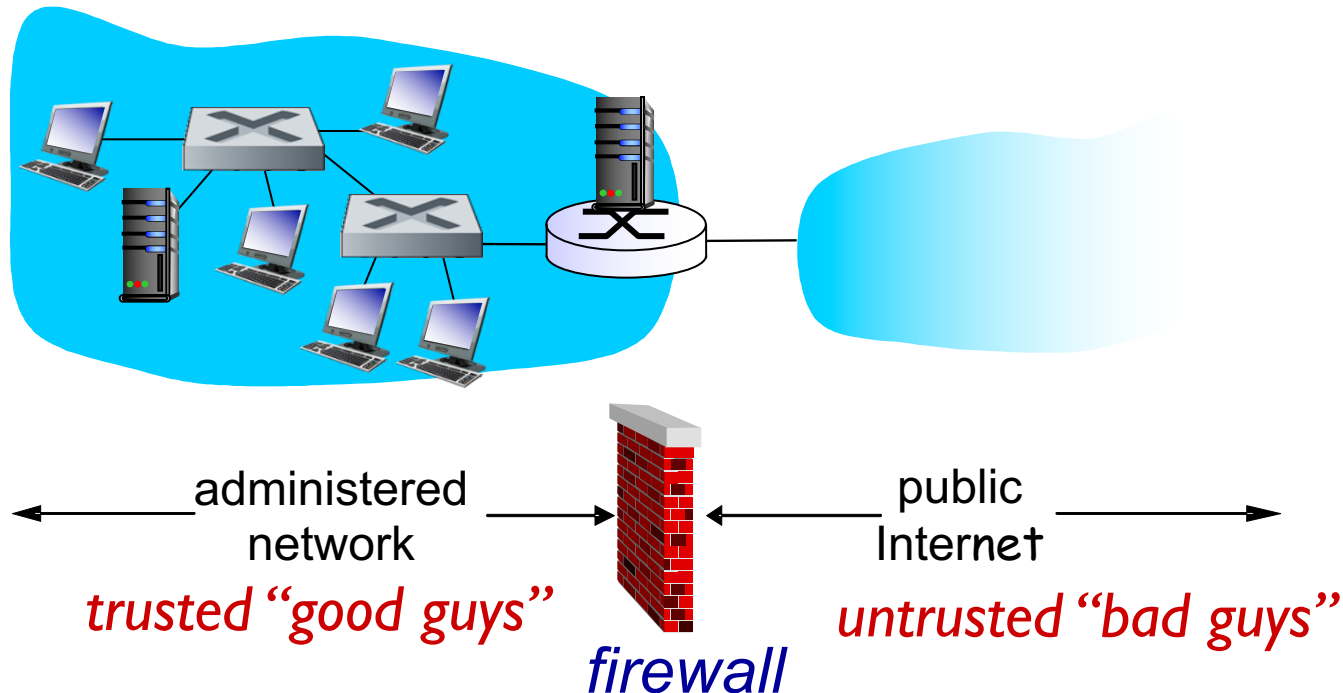
~~8.7 Securing wireless LANs~~

*8.8 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

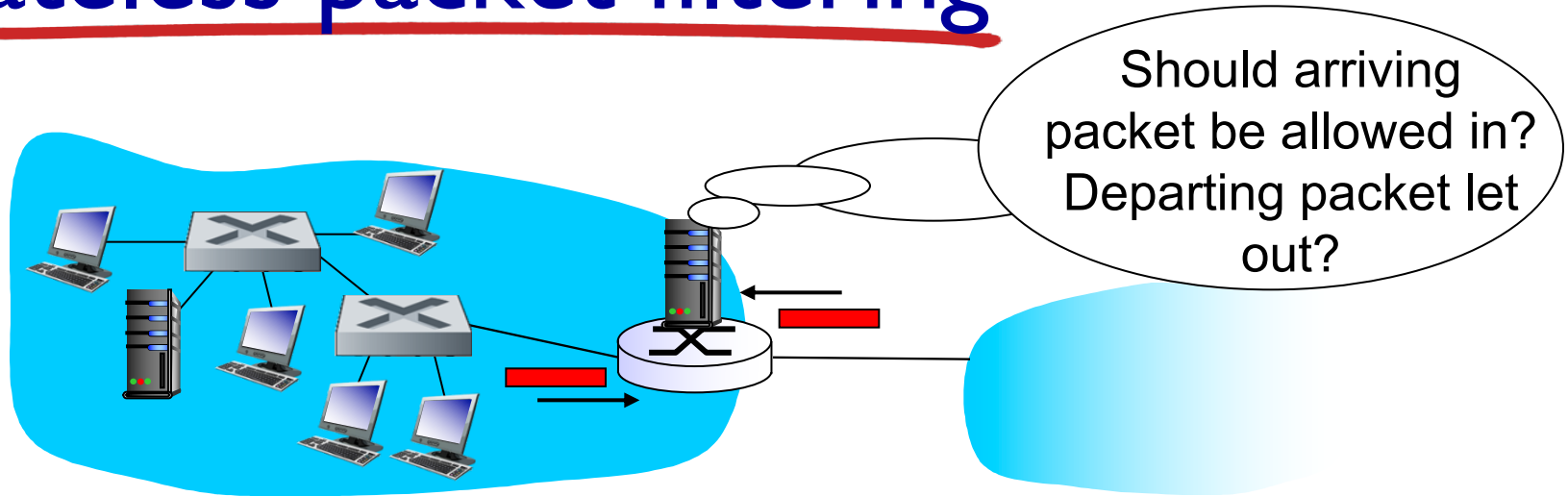
allow only authorized access to inside network

- 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: looks like OpenFlow forwarding (Ch. 4)!

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



# 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	

# 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

# Network Security (summary)

## basic techniques.....

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

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

- secure transport (SSL)
- IP sec

## operational security: firewalls and IDS